Advances in Engineering Software, Vol.123, pp.117-129, 2018, Elsevier ISSN 0965-9978

Vector Field Radial Basis Function Approximation

Michal Smolik[a,*], Vaclav Skala[a], Zuzana Majdisova[a]

[a] *Faculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic*

Abstract

Vector field simplification aims to reduce the complexity of the flow by removing features according to their relevance and importance. Our goal is to preserve only the important critical points in the vector field and thus simplify the vector field for the visualization purposes. We use Radial Basis Functions (RBF) approximation with Lagrange multipliers for vector field approximation. The proposed method was experimentally verified on synthetic and real weather forecast data sets. The results proved the quality of the proposed approximation method compared to other existing approaches. A significant contribution of the proposed method is an analytical form of the vector field which can be used in further processing.

Keywords: Vector field, Radial basis functions, critical point, Lagrange multipliers, simplification, Fourier transform, Cosine transform
2010 MSC: 00-01, 99-00

1. Introduction

Interpolation and approximation are probably the most frequent operations used in computational techniques [1]. Several techniques have been developed for data interpolation and approximation, but they require some kind of data "ordering", e.g. structured mesh, rectangular mesh, unstructured mesh etc. A typical example is a solution of partial differential equations (PDE), where derivatives are replaced by differences and rectangular or hexagonal meshes are used in the vast majority of cases. However, in many engineering problems, data are not ordered and they are scattered in k-dimensional space, in general. The k-dimensional space is sometimes

☆The research was supported by projects Czech Science Foundation (GACR) No. 17-05534S and partially by SGS 2016-013.

*Corresponding author
. E-mail address: smolik@kiv.zcu.cz (M. Smolik)

https://doi.org/10.1016/j.advengsoft.2018.06.013

not only spatial but also contains a time dimension or a dimension relating to age or temperature or other environmental conditions. Usually, in technical applications the scattered data are tessellated using triangulation, but this approach is quite prohibitive for the case of k-dimensional data interpolation or approximation because of the computational cost [2].

The technique for visualizing topological information in fluid flows is well known [3]. However, when the technique is used in complex and information-rich data sets, the result will be a cluttered image which is difficult to interpret. The paper [4] presents a simplification approach that removes pairs of critical points from the dataset, based on relevance measures. The approach does no grid changes since the whole method uses small local changes of the vector values defining the vector field. A simplification of vector field can be achieved by merging critical points within a prescribed radius into higher order critical points [5]. After building clusters containing the singularities to merge, the method generates a piecewise linear representation of the vector field in each cluster containing only one higher order singularity. Paper [6] presents a method to segment regions around a higher order critical point into areas of different $3D$ flow behavior. This method can be applied to any area of interest, e.g. around clusters of critical points. This can be used for a topological simplification tool by replacing the topological skeleton inside the area of interest. Combination of topological simplification technique and topology preserving compression for $2D$ vector fields is presented in [7]. A vector field is compressed in such way that its important topological features are preserved while its unimportant features are allowed to collapse and disappear. [8] presents a Delaunay based algorithm for simplifying vector fields. The algorithm controls a local metric during removing vertices from Delaunay triangulation and maintains regions near critical points to prevent topological changes. The paper [9] uses a filtering technique based on the vorticity of the vector field to eliminate the less interesting critical points. The magnitude of the curl of the scalar field provides a basis to control the boundary thresholds as well as the number of critical points to include in the vector field. The paper [10] presents a technique for the visualization of multi-level topology in flow data sets. It provides the user with a mechanism to visualize the topology without excessive cluttering while maintaining the global structure of the flow. [11] and [12] enable the pruning of sets of critical points according to a quantitative measure of their stability, that is, the minimum amount of vector field perturbation required to remove them. This leads to a hierarchical simplification scheme that encodes flow magnitude in its perturbation metric. A topological denoising technique based on a global energy optimization is proposed in [13], which allows the topology-controlled denoising of scalar fields. It allows processing small patches of the domain indepen-

https://doi.org/10.1016/j.advengsoft.2018.06.013

dently while still avoiding the introduction of new critical points. In the paper [14], they performed a numerical investigation of the differences between RBF global and local methods, in order to investigate the possible advantage of using local methods for the approximation of vector fields. The paper [15] presents a vector field approximation for two-dimensional vector fields that preserves their topology and significantly reduces the memory footprint. This approximation is based on a segmentation. The flow within each segmentation region is approximated by an affine linear function.

2. Vector Field

Vector fields on surfaces are important objects, which appear frequently in scientific simulation in CFD (Computational Fluid Dynamics) [16], [17] or modeling by FEM (Finite Element Method) [18], [19]. To be visualized, such vector fields are usually linearly approximated for the sake of simplicity and performance considerations.

The vector field can be easily analyzed when having an approximation of the vector field near some location point. The important places to be analyzed are so called critical points [20]. Analyzing the vector field behavior near these points gives us the information about the characteristic of the vector field.

Critical points $\boldsymbol{x}_0$ of the vector field are points at which the magnitude of the vector vanishes

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}(\boldsymbol{x}) = \mathbf{0}, \tag{1}$$

i.e. all components are equal to zero

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2}$$

A critical point is said to be isolated, or simple, if the vector field is non vanishing in an open neighborhood around the critical point. Thus for all surrounding points $\boldsymbol{x}_\epsilon$ of the critical point $\boldsymbol{x}_0$ the equation (1) does not apply, i.e.

$$\frac{d\boldsymbol{x}_\epsilon}{dt} \neq \mathbf{0}, \tag{3}$$

At critical points, the direction of the field line is indeterminate, and they are the only points in the vector field where field lines can intersect (asymptotically). The terms singular point, null point, neutral point or equilibrium point are also frequently used to describe critical points.

https://doi.org/10.1016/j.advengsoft.2018.06.013

These points are important because together with the nearby surrounding vectors, they have more information encoded in them than any such group in the vector field, regarding the total behavior of the field. The critical points are classified based on the vector field around these points, see Figure 1.

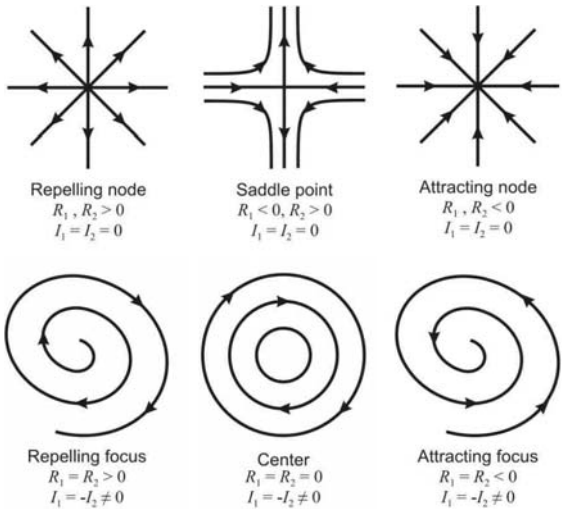


Figure 1: Classification of $2D$ first order critical points. R_1, R_2 denote the real parts of the eigenvalues of the Jacobian matrix while I_1, I_2 denote their imaginary parts [20].

3. Radial Basis Functions

Radial basis function (RBF) is a technique for scattered data interpolation [21] and approximation [22], [23]. The RBF interpolation and approximation is computationally more expensive compared to interpolation and approximation methods that use an information about mesh connectivity, because input data are not ordered and there is no known relation between them, i.e. tessellation is not made. Although RBF has a higher computational cost, it can be used for d-dimensional problem solution in many applications, e.g. solution of partial differential equations [24], [25], image reconstruction [26], neural networks [27], [28], [29], GIS systems [30], [31], optics [32] etc. It should be noted that it does not require any triangulation or tessellation mesh in general. There is no need to know any connectivity of interpolation points, all points are tied up only with distances of each other. Using all these distances we can form the interpolation or approximation matrix, which will be shown later.

https://doi.org/10.1016/j.advengsoft.2018.06.013

The RBF is a function whose value depends only on the distance from its center point. Due to the use of distance functions, the RBFs can be easily implemented to reconstruct the surface using scattered data in 2D, 3D or higher dimensional spaces. It should be noted that the RBF interpolation and approximation is not separable by dimension.

Radial function interpolants have a helpful property of being invariant under all Euclidean transformations, i.e. translations, rotations and reflections. It does not matter whether we first compute the RBF interpolation function and then apply a Euclidean transformation, or if we first transform all the data and then compute the radial function interpolants. This is a result of the fact that Euclidean transformations are characterized by orthonormal transformation matrices and are therefore two-norm invariant. Radial basis functions can be divided into two groups according to their influence. The first group are "global" RBFs [33]. Application of global RBFs usually leads to ill-conditioned system, especially in the case of large data sets with a large span [34], [35].

The "local" RBFs were introduced in [36] as compactly supported RBF (CSRBF) and satisfy the following condition:

$$\begin{aligned}\varphi(r) &= (1-r)_+^q P(r) \\ &= \begin{cases}(1-r)^q P(r) & 0 \le r \le 1 \\ 0 & r > 1\end{cases}\end{aligned} \tag{4}$$

where $P(r)$ is a polynomial function, r is the distance of two points and q is a parameter. The subscript in $(1-r)_+^q$ means:

$$(1-r)_+ = \begin{cases}(1-r) & (1-r) \ge 0 \\ 0 & (1-r) < 0\end{cases} \tag{5}$$

3.1. Radial Basis Function Approximation

RBF interpolation was originally introduced by [37] and is based on computing the distance of two points in any k-dimensional space. The interpolated value, and approximated value as well, is determined as (see [38]):

$$h(\boldsymbol{x}) = \sum_{j=1}^{M} \lambda_j \varphi(\|\boldsymbol{x} - \boldsymbol{\xi}_j\|) \tag{6}$$

where λ_j are weights of the RBFs, M is the number of the radial basis functions, φ is the radial basis function and $\boldsymbol{\xi}_j$ are centers of radial basis functions. For a given

https://doi.org/10.1016/j.advengsoft.2018.06.013

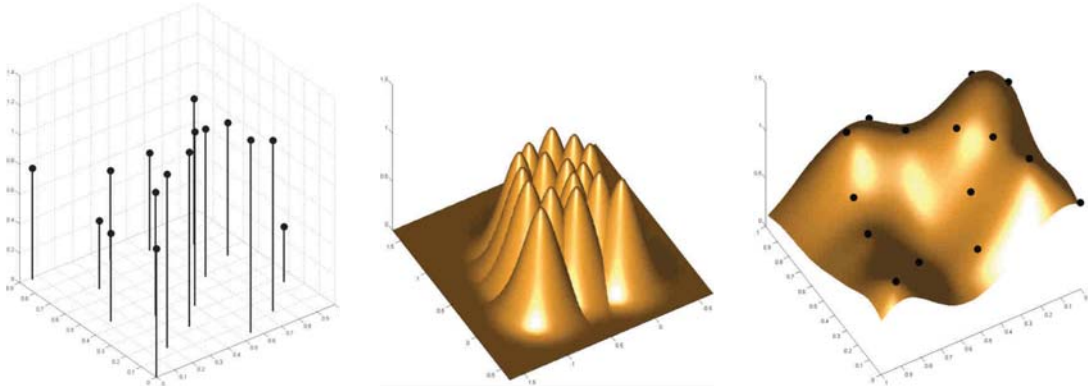


Figure 2: Data values, the RBF collocation functions, the resulting interpolant.

dataset of points with associated values, i.e. in the case of scalar values $\{\boldsymbol{x}_i, h_i\}_1^N$, where $N \gg M$, the following overdetermined linear system of equations is obtained:

$$h_i = h(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j \varphi(\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|) \quad \text{for } \forall i \in \{1, \ldots, N\} \quad (7)$$

where λ_j are weights to be computed; see Figure 2 for a visual interpretation of (6) or (7) for a $2\frac{1}{2}D$ function. Point in $2\frac{1}{2}D$ is a $2D$ point associated with a scalar value. The same also applies to $3D$ point associated with a scalar value, thus $3\frac{1}{2}D$ point.

Equation (7) can be rewritten in a matrix form as

$$\boldsymbol{A}\boldsymbol{\lambda} = \boldsymbol{h}, \quad (8)$$

where $A_{ij} = \varphi(\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|)$ is the entry of the matrix in the $i-$th row and $j-$th column, the number of rows $N \gg M$, M is the number of unknown weights $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_M]^T$, i.e. a number of reference points, and $\boldsymbol{h} = [h_1, \ldots, h_N]^T$ is a vector of values in the given points. The presented system is overdetermined, i.e. the number of equations N is higher than the number of variables M. This linear system of equations can be solved by the least squares method (LSE) as

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\lambda} = \boldsymbol{A}^T\boldsymbol{h}, \quad (9)$$

where the matrix $\boldsymbol{A}^T\boldsymbol{A}$ is symmetrical.

https://doi.org/10.1016/j.advengsoft.2018.06.013

The RBF approximation can be done using "global" or "local" functions. When using "global" radial basis functions, the matrix $\boldsymbol{A}$ will be full, but when using "local" radial basis functions, the matrix $\boldsymbol{A}$ might be sparse, which can be beneficial when solving the overdetermined system of linear equations $\boldsymbol{A\lambda} = \boldsymbol{h}$.

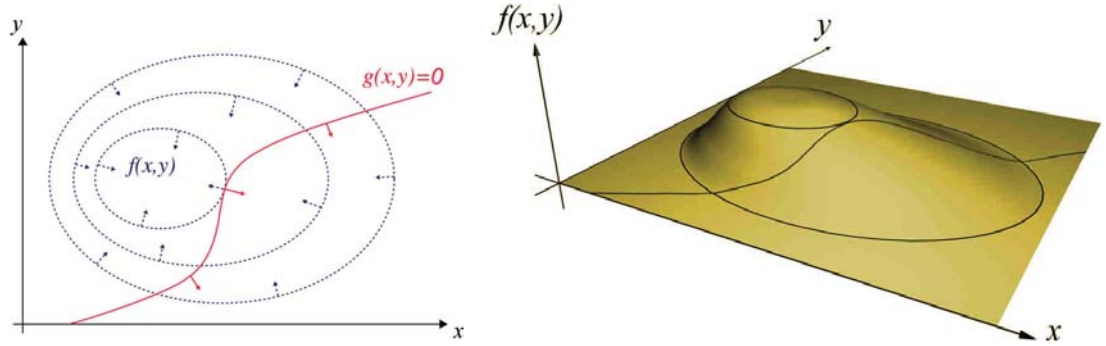
In the case of the vector data, i.e. $\{\boldsymbol{x}_i, \boldsymbol{h}_i\}_1^N$ values $\boldsymbol{h}_i$ are actually vectors, the RBF is to be performed for each coordinate of the vector $\boldsymbol{h}_i$.

3.2. RBF approximation with Lagrange multiplier

We want to minimize a multivariate function $f(x, y)$ subject to a constraint $g(x, y) = 0$. The method of Lagrange multipliers relies on the intuition that at a minimum, $f(x, y)$ cannot be decreasing in the direction of any neighboring point where $g(x, y) = 0$. Thus, the gradient of $f(x, y)$ is perpendicular to the constraint $g(x, y) = 0$ and thus the gradients of f and g are parallel, see Figure 3, i.e.

$$\nabla f\,(x, y) = \eta \nabla g\,(x, y)\,, \tag{10}$$

where η represents some constant.



(a) $2D$ view. (b) $3D$ view.

Figure 3: The red line shows the constraint $g(x, y) = 0$. The blue lines are contours of $f(x, y)$. The point where the red line tangentially touches a blue contour is the maximum of $f(x, y)$.

The method of Lagrange multipliers is a powerful tool for solving this class of problems without the need to explicitly solve the conditions and use them to eliminate extra variables. Lagrange presented a special new function which takes in all the same input variables as f and g, along with η, thought of now as a variable called Lagrange multiplier. The Lagrange function is [39], [40], [41]

$$F(x, y, \eta) = f(x, y) + \eta g(x, y). \tag{11}$$

https://doi.org/10.1016/j.advengsoft.2018.06.013

To find a minimum of (11) we need to set the gradient of (11) equal to zero

$$\begin{aligned} \frac{\partial F}{\partial \boldsymbol{x}} &= \mathbf{0} \\ \frac{\partial F}{\partial \eta} &= 0, \end{aligned} \tag{12}$$

where $\boldsymbol{x} = [x, y]^T$.

We will use Lagrange multipliers together with RBF approximation. The RBF approximation function is

$$h(\boldsymbol{x}) = \sum_{j=1}^{M} \lambda_j \varphi \left(\| \boldsymbol{x} - \boldsymbol{\xi}_j \| \right), \tag{13}$$

where $\boldsymbol{\xi}_j$ are centers of radial basis functions, λ_j are weights of radial basis functions and φ is a radial basis function. We want to minimize

$$f(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \left(\sum_{j=1}^{M} \lambda_j \varphi \left(\| \boldsymbol{x}_i - \boldsymbol{\xi}_j \| \right) - h_i \right)^2, \tag{14}$$

where N is the number of input points for approximation and h_i are associated function values at $\boldsymbol{x}_i$. The constrain for some points $\boldsymbol{x^{(0)}} = \left[\boldsymbol{x}_1^{(\mathbf{0})}, \ldots, \boldsymbol{x}_{N_c}^{(\mathbf{0})} \right]$ that the RBF function is equal to zero is

$$g(\boldsymbol{x}_c^{(\mathbf{0})}) = \sum_{j=1}^{M} \lambda_j \varphi \left(\left\| \boldsymbol{x}_c^{(\mathbf{0})} - \boldsymbol{\xi}_j \right\| \right) = 0 \qquad \text{for } \forall c \in \{1, \ldots, N_c\}, \tag{15}$$

where N_c is the number of constrains.

Using (14) as $f(x, y)$ in (11) and (15) as $g(x, y)$ in (11) we form the Lagrange function

$$F(\boldsymbol{\lambda}, \boldsymbol{\eta}) = \sum_{i=1}^{N} \left(\sum_{j=1}^{M} \lambda_j \varphi \left(\| \boldsymbol{x}_i - \boldsymbol{\xi}_j \| \right) - h_i \right)^2 + \sum_{c=1}^{N_c} \eta_c \left(\sum_{j=1}^{M} \lambda_j \varphi \left(\left\| \boldsymbol{x}_c^{(\mathbf{0})} - \boldsymbol{\xi}_j \right\| \right) \right). \tag{16}$$

This formula can be rewritten in a matrix form as

$$\begin{aligned} F(\boldsymbol{\lambda}, \boldsymbol{\eta}) &= (\boldsymbol{A}\boldsymbol{\lambda} - \boldsymbol{h})^2 + \boldsymbol{\lambda}^T \boldsymbol{R}^T \boldsymbol{\eta} \\ &= \boldsymbol{\lambda}^T \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{\lambda} - 2\boldsymbol{h}^T \boldsymbol{A} \boldsymbol{\lambda} + \boldsymbol{h}^T \boldsymbol{h} + \boldsymbol{\lambda}^T \boldsymbol{R}^T \boldsymbol{\eta}, \end{aligned} \tag{17}$$

https://doi.org/10.1016/j.advengsoft.2018.06.013

where $A_{ij} = \varphi\left(\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|\right)$ is the entry of the matrix $\boldsymbol{A}$ in the i-th row and the j-th column, $R_{cj} = \varphi\left(\left\|\boldsymbol{x}_c^{(\mathbf{0})} - \boldsymbol{\xi}_j\right\|\right)$ is the entry of the matrix $\boldsymbol{R}$, $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_M]^T$, $\boldsymbol{h} = [h_1, \ldots, h_N]^T$ and $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_{N_c}]^T$. To find the minimum of (17) we need to compute the partial derivatives of (17)

$$\frac{\partial F}{\partial \boldsymbol{\lambda}} = 2\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\lambda} + \boldsymbol{R}^T\boldsymbol{\eta} - 2\boldsymbol{A}^T\boldsymbol{h} \tag{18}$$

and

$$\frac{\partial F}{\partial \boldsymbol{\eta}} = \boldsymbol{R}\boldsymbol{\lambda}. \tag{19}$$

To find the minimum, both partial derivatives (18), (19) must be equal zero

$$\begin{aligned} &\frac{\partial F}{\partial \boldsymbol{\lambda}} = \mathbf{0} \Rightarrow 2\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\lambda} + \boldsymbol{R}^T\boldsymbol{\eta} - 2\boldsymbol{A}^T\boldsymbol{h} = \mathbf{0}, \\ &\frac{\partial F}{\partial \boldsymbol{\eta}} = \mathbf{0} \Rightarrow \boldsymbol{R}\boldsymbol{\lambda} = \mathbf{0}. \end{aligned} \tag{20}$$

Using (20), we can form the system of linear equations for RBF approximation with Lagrange multipliers

$$\left[\begin{array}{c|c} 2\boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{R}^T \\ \hline \boldsymbol{R} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\lambda} \\ \hline \boldsymbol{\eta} \end{array}\right] = \left[\begin{array}{c} 2\boldsymbol{A}^T\boldsymbol{h} \\ \hline \mathbf{0} \end{array}\right] \tag{21}$$

Solving those equations and finding $\boldsymbol{\lambda}$, we made the RBF approximation and can compute the function values using (13).

4. Proposed Approach

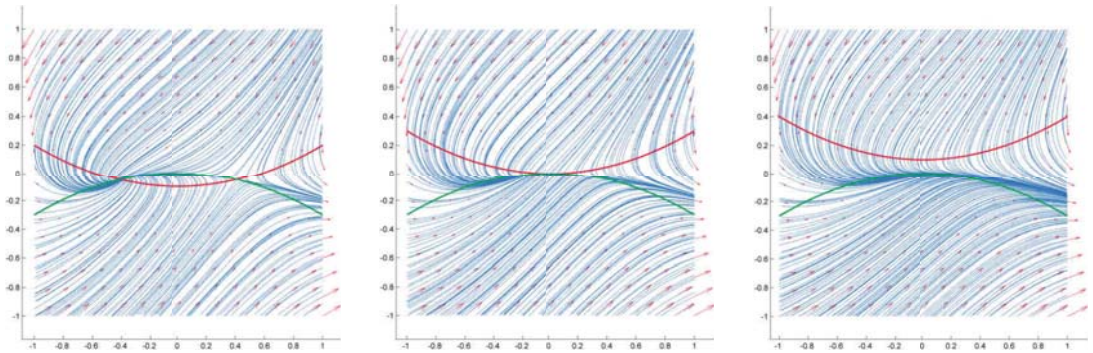
Importance measure for minima and maxima of $2D$ scalar fields called scale space persistence is introduced in [42]. This method is based on the mathematical concepts scale space, homological persistence, discrete Morse theory, and combinatorial feature flow fields. The combination of these powerful approaches to data analysis results in an importance measure that can deal with noisy data containing outliers. Robustness, a notion related to persistence, is used to represent the stability of critical points and evaluate their significance with respect to perturbations of the vector field. Intuitively, the robustness of a critical point is the minimum amount of perturbation that is required to cancel this critical point within a local neighborhood.

The proposed algorithm aims to reduce critical points in the vector field and thus simplify it. Critical points are located on places where the zero iso-lines of v_x and v_y

https://doi.org/10.1016/j.advengsoft.2018.06.013

intersect. The importance of the critical point is based on how the two zero iso-lines of v_x and v_y intersect and how long are these iso-lines.

If two iso-lines are almost parallel at a critical point, then this critical point does not have high importance as can be seen in the synthetic example in Figure 4. The two zero iso-lines have two intersections (Figure 4a), thus the vector field has two critical points at that locations. If we change the vector field in Figure 4a so that the vector field has only one critical point and both the v_x and v_y zero iso-lines touches at one point, i.e. at critical point, see Figure 4b. If we change the vector field Figure 4a even more, we can end up with the vector field Figure 4c, which has no critical point. All the vector fields Figure 4a-c have the same global character and differ only in the number of critical points. Due to inaccuracy in measurement or numerical simulation or vector field data noise, there may be confusion between the individual cases in Figure 4a-c. Some non existing critical points can occur in the data set and are disturbing for visualization purposes.



(a) (b) (c)

Figure 4: Vector fields with almost parallel zero iso-lines. All the vector fields have the same global character.

If two iso-lines have an intersection, where one iso-line is a relatively short closed curve, the influence of the critical point is only local and has almost no global influence for the vector field. Examples are visualized in Figure 5 and Figure 6. We can change the vector field Figure 5 (left) that the v_y component of the vector field is always greater than zero, i.e. there is no zero iso-line, see Figure 5 (right). However this modified vector field has no critical point, the global character of the vector field remains the same as in Figure 5 (left). The same implies for vector field in Figure 6 as well. Critical points created with a relatively short closed curve have the influence proportional to the size of the zero iso-line.

https://doi.org/10.1016/j.advengsoft.2018.06.013

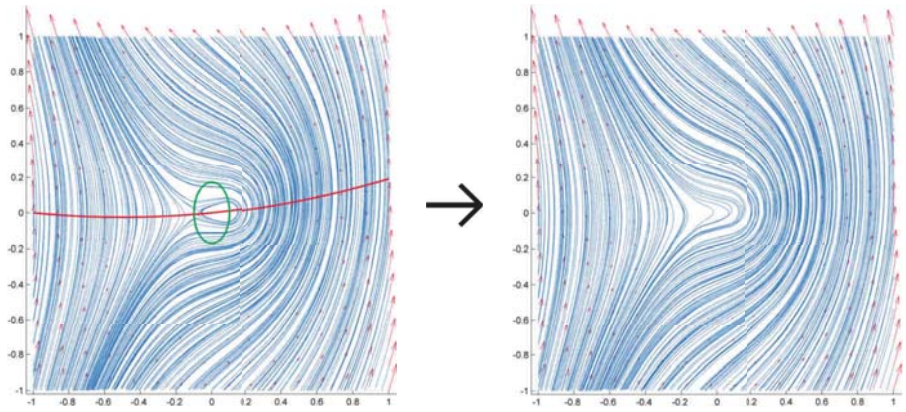


Figure 5: Vector fields with one short zero iso-lines. Both the vector fields have the same global character.

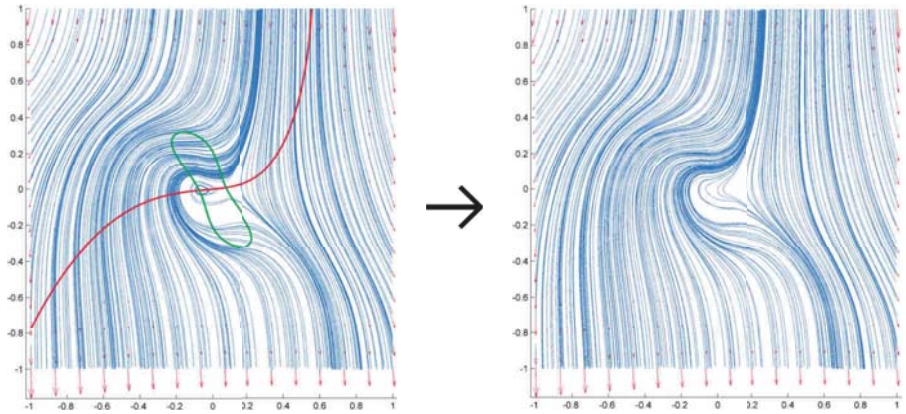


Figure 6: Vector fields with one short zero iso-lines. Both the vector fields have the same global character.

It can be seen that for the both cases mentioned above, i.e. almost parallel iso-lines in Figure 4 and relatively short closed curve in Figure 5 and Figure 6, the vector fields have the same global character with the critical points as without them. In the following text we will show how to detect critical points affecting only local behavior of the vector field.

4.1. Critical points reduction

The first step of the proposed method is the critical points detection. Critical points in the vector field can be located using several methods [43], [44], [45]. Critical

https://doi.org/10.1016/j.advengsoft.2018.06.013

points of the $2D$ vector field are places, where the two zero iso-lines of v_x and v_y component intersect. Our goal is to trace both zero iso-lines in both directions from the critical point and compute the approximated zero iso-lines.

For the v_x zero iso-line approximation we need to compute intersections with the input data grid. From these intersections, we find some number of the closest ones to the critical point $\boldsymbol{x}_0$. In the testing we used 6 closest intersection points $\left\{\boldsymbol{x}_c^{(1)}, \ldots, \boldsymbol{x}_c^{(6)}\right\}$. This points need to be interlaced with an implicit line. For this purpose we use the method of total least square error computation in E^2 taken from [46]. This method computes implicit line

$$a^{(x)}x + b^{(x)}y + d^{(x)} = 0, \tag{22}$$

where $a^{(x)}$, $b^{(x)}$ and $d^{(x)}$ are parameters of implicit line for v_x, similarly $a^{(y)}$, $b^{(y)}$ and $d^{(y)}$ for v_y. This line has a normal vector

$$\boldsymbol{n}^{(x)} = [a^{(x)}, b^{(x)}]^T, \tag{23}$$

and thus the direction vector of the implicit line is

$$\boldsymbol{u}^{(x)} = [b^{(x)}, -a^{(x)}]^T. \tag{24}$$

The direction vector is a smooth approximation of the zero iso-line, see Figure 7. For the v_y zero iso-line smooth approximation, we compute $\boldsymbol{u}^{(y)}$ using the same approach like for $\boldsymbol{u}^{(x)}$.

If two zero iso-lines are almost parallel (as in Figure 4), the two corresponding direction vectors $\boldsymbol{u}^{(x)}$ and $\boldsymbol{u}^{(y)}$ computed using (24) are almost parallel as well. The angle θ between this two vectors is computed using

$$\theta = acos\left(\frac{\boldsymbol{u}^{(x)} \cdot \boldsymbol{u}^{(y)}}{\|\boldsymbol{u}^{(x)}\| \, \|\boldsymbol{u}^{(y)}\|}\right). \tag{25}$$

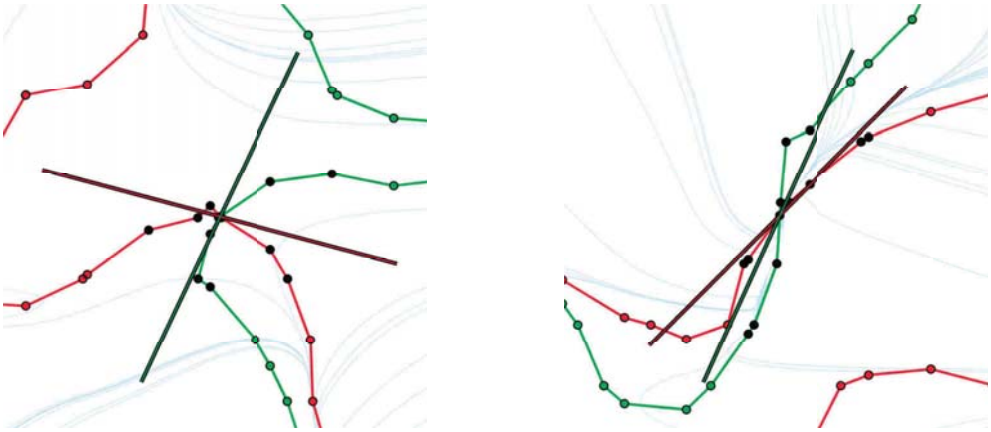
To distinguish cases, when vectors are almost parallel and when not, we can compute only the argument of function *acos* in (25)

$$\nu = \frac{\boldsymbol{u}^{(x)} \cdot \boldsymbol{u}^{(y)}}{\|\boldsymbol{u}^{(x)}\| \, \|\boldsymbol{u}^{(y)}\|}. \tag{26}$$

The two vectors $\boldsymbol{u}^{(x)}$ and $\boldsymbol{u}^{(y)}$ are almost parallel, when

$$|\nu| \longrightarrow 1. \tag{27}$$

https://doi.org/10.1016/j.advengsoft.2018.06.013



(a) (b)

Figure 7: Examples of zero iso-line approximation. The black line is the zero iso-line approximation at the critical point. The red and green curves are zero iso-lines.

In our approach we are going to remove all critical points for which the condition (27) is true for some small ϵ in

$$|\nu| \geq 1 - \epsilon. \tag{28}$$

This critical points have only local character and are not important for the global character of the vector field.

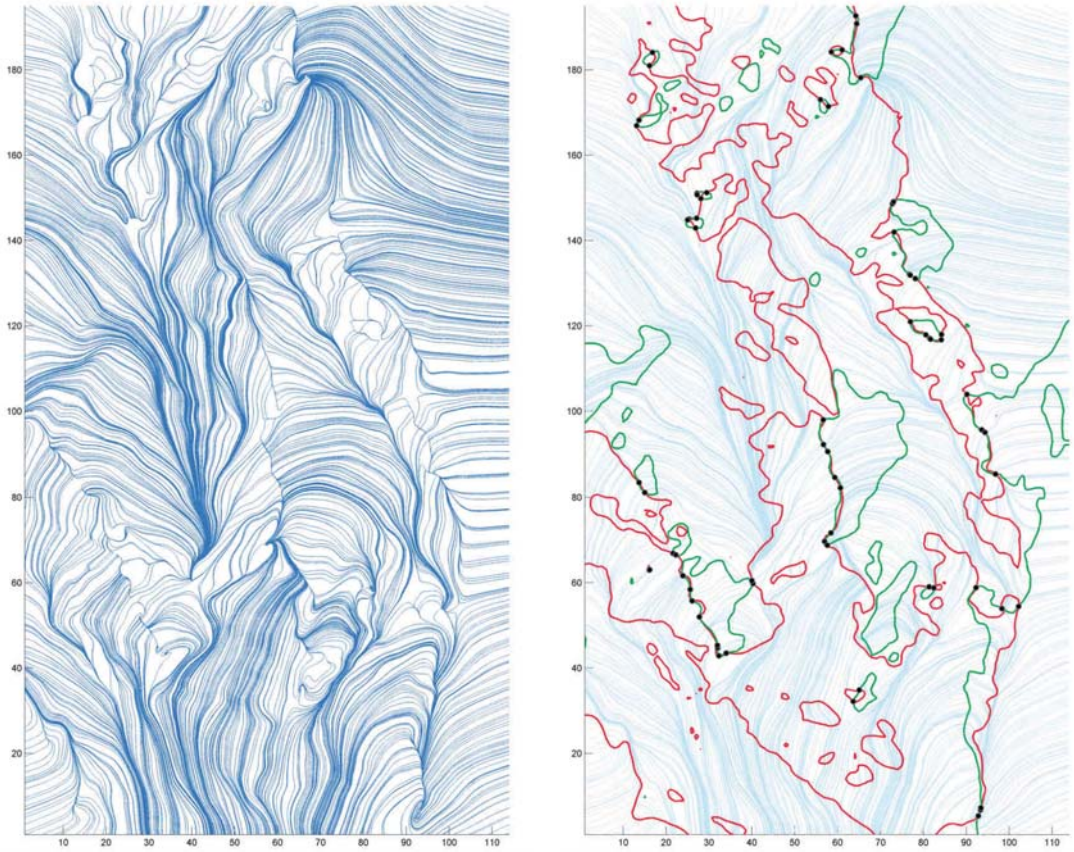
The next type of critical points that have only local character is described in Figure 5 and Figure 6. To detect this type of critical points, we need to trace the zero iso-line of v_x and v_y and compute how long this zero iso-lines are. If one zero iso-line is shorter than some threshold, then the critical point have only local character and can be removed. The threshold length needs to be determined experimentally relatively to the size of input data set.

4.2. Vector field RBF approximation

Now, we know the critical points that can be removed from the vector field as they change the vector field only locally and thus have no importance for the global character of the vector field. The remaining critical points need to be preserved in the approximated vector field. The vector field will be approximated using the RBF with Lagrange multipliers as described in Section (3.2), so that the important critical points will be preserved in the approximated vector field. For the RBF approximation, we need to determine the centers of radial basis functions.

Centers of the RBF approximation need to be located in the positions of the important critical points. The additional centers of RBF need to be located at the

https://doi.org/10.1016/j.advengsoft.2018.06.013



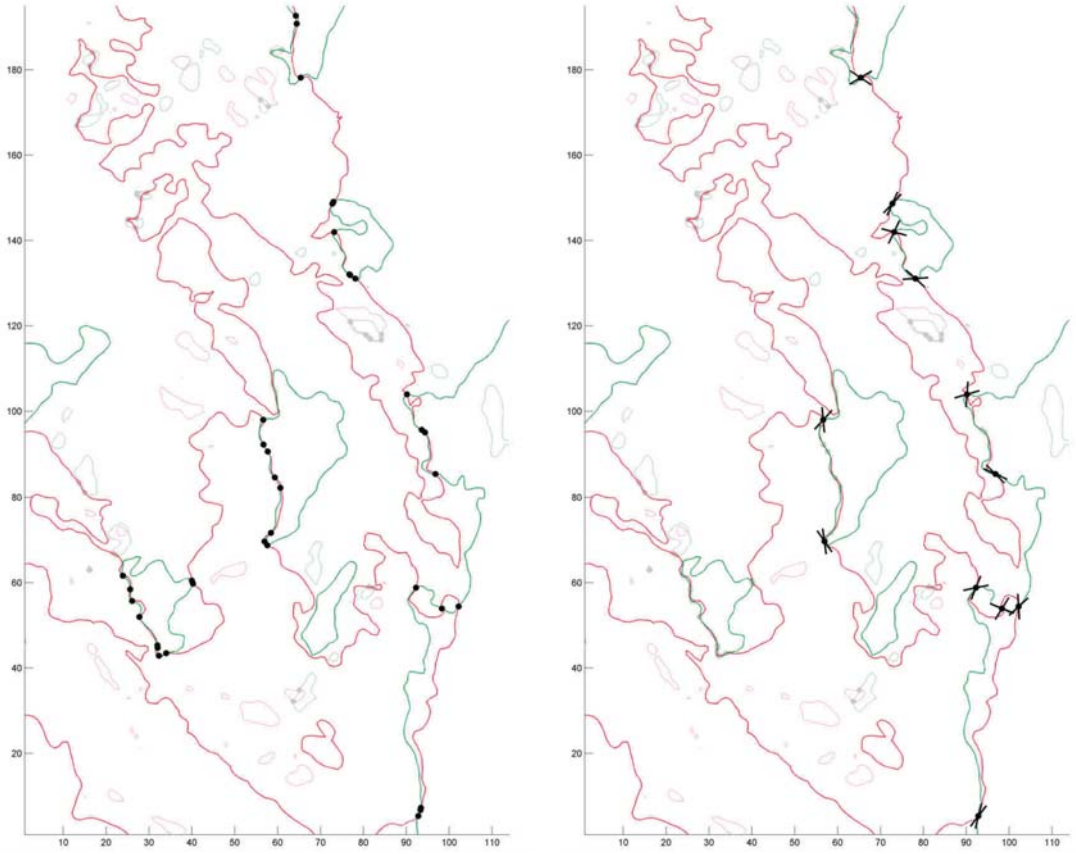
(a) Vector field. (b) Zero iso-lines and critical points.

Figure 8: Visualization of the input data set. The vector field (a) and the zero iso-lines with all critical points (b).

extremes of v_x, resp. v_y. The number of extremes will be too high for real data due to noise, measurement inaccuracies and calculation. Therefore, the v_x, resp. v_y, component of the vector field is smoothed using Gaussian low-pass filter before computing extremes. Now, we can compute the RBF approximation with Lagrange multiplier as described in section 3.2. The conditions for Lagrange multiplier are the zero values at critical points locations.

After computing RBF approximation, we have an analytical description of the vector field. This approximated vector field can be used for visualization purposes as the global character of the vector field remains the same and only small local changes are neglected.

https://doi.org/10.1016/j.advengsoft.2018.06.013



(a) (b)

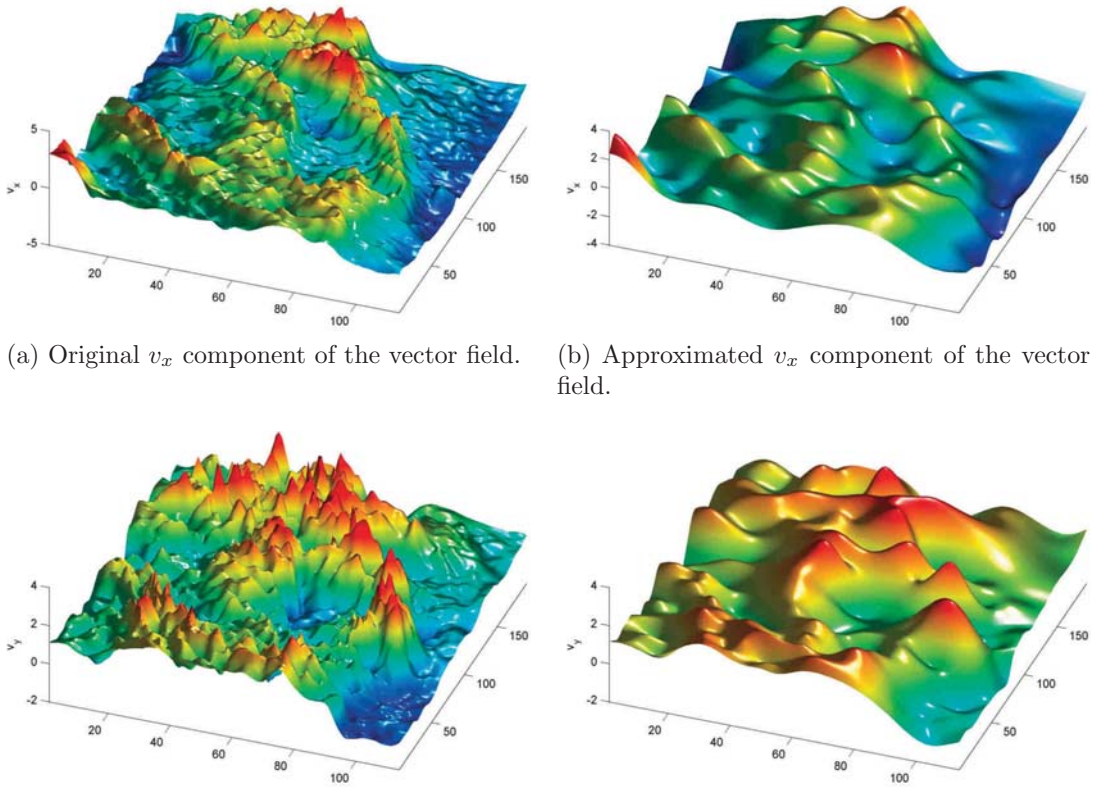
Figure 9: First (a) and second (b) step of critical points reduction algorithm.

5. Results

In this section, we show the results of our proposed approach. This approach for vector field approximation using radial basis functions is especially convenient for visualization purposes, data understanding and data compression. For testing purposes, we used a numerical forecast wind data set taken from[1] [47]. This data set consists of $2.2 \cdot 10^4$ sampled points with associated vectors of wind flow, see Figure 8a. The vector field contains 69 critical points as can be seen in Figure 8b, but only few

[1]Data set of wind flow at a height of 10m over the surface of the Czech Republic courtesy of the Institute of Computer Science of the Czech Academy of Sciences.

https://doi.org/10.1016/j.advengsoft.2018.06.013



(a) Original v_x component of the vector field.

(b) Approximated v_x component of the vector field.

(c) Original v_y component of the vector field.

(d) Approximated v_y component of the vector field.

Figure 10: Original and RBF approximated components of vector field.

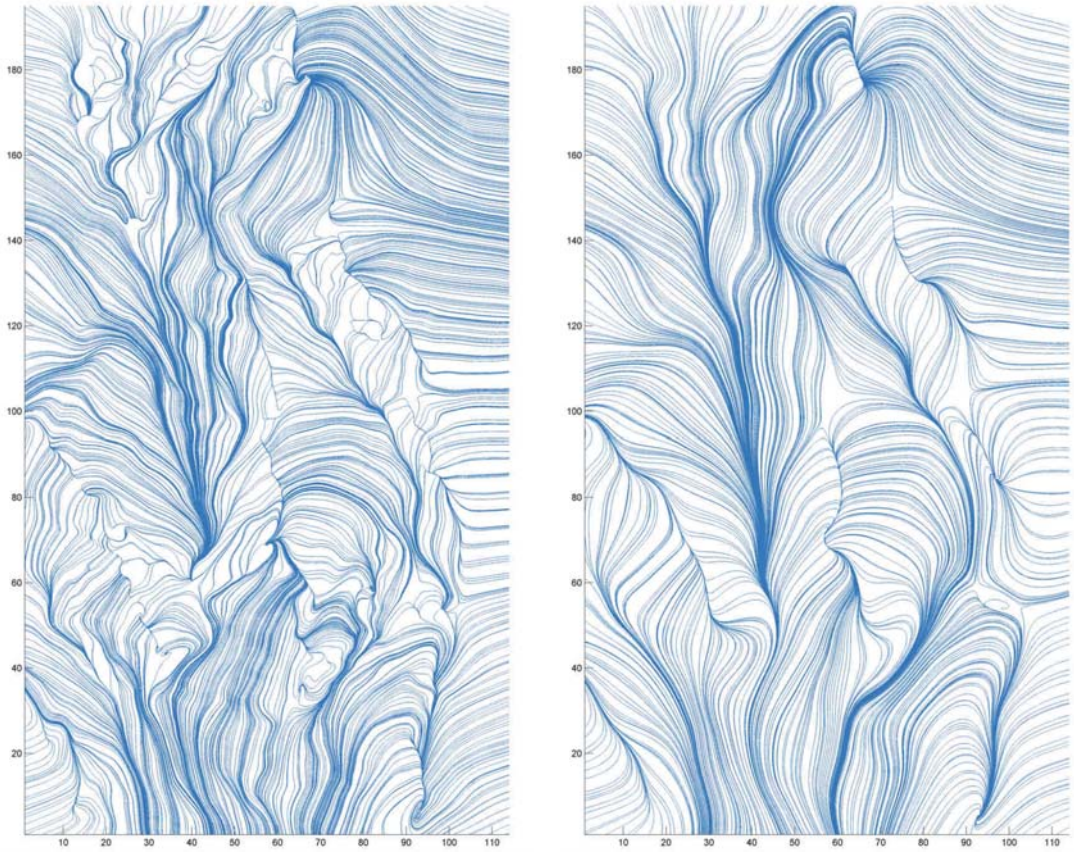
of then are important and have the global character.

We need to select only the important critical points and eliminate the others. First set of critical points, that can be eliminated, are critical points laying on a relatively short iso-lines. For processing only critical points with longer iso-lines length are considered. The threshold of zero iso-line length was experimentally selected as half of the minimum of x and y axis range, i.e.

$$\frac{min\left\{range_x, range_y\right\}}{2}. \tag{29}$$

After ignoring this short zero iso-lines, we obtain reduced set of critical points, see Figure 9a. To eliminate the second group of critical points, we need to compute (26)

https://doi.org/10.1016/j.advengsoft.2018.06.013



(a) Original vector field. (b) Approximated vector field.

Figure 11: Original vector field (a) and approximated vector field using the proposed approach (b).

and compare it to some threshold value ϵ in (28). Experimentally, we selected the value of $\epsilon = 0.06$, which is relevant to an angle equal 20°, i.e. $\cos^{-1}(1-\epsilon) = 20°$. As the result we end up with only 12 critical points, this means that we eliminated 57 critical points, see Figure 9b.

Both the vector field components v_x and v_y were approximated by RBF using the Lagrange multiplier. The result of approximation is visualized at Figure 10. It can be seen that both approximations have more smoothed course of the function compared to the original functions. This property is beneficial for the visualization purposes as the final vector field does not contain small fluctuations.

Having approximation of both vector field components v_x and v_y, we can visualize the approximated vector field, see Figure 11b and Figure 12. It can be seen that

https://doi.org/10.1016/j.advengsoft.2018.06.013

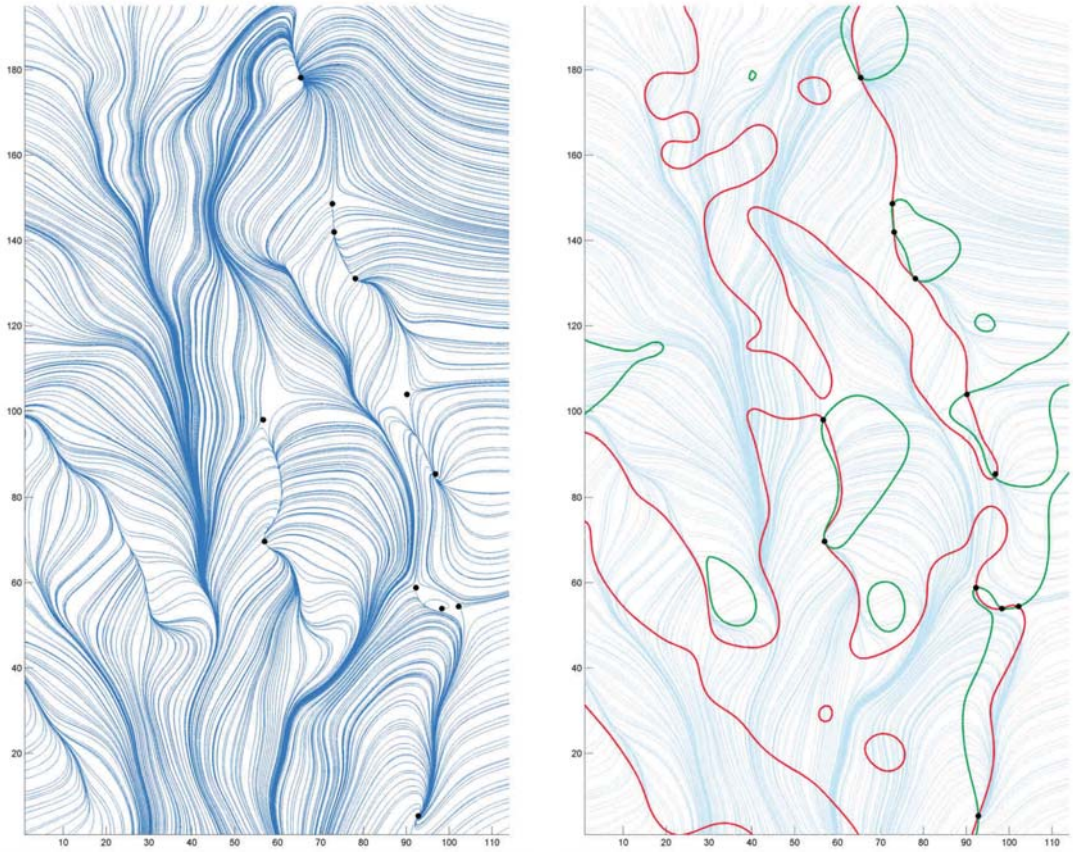


Figure 12: Approximated vector field with its all critical points (a) and zero iso-lines of the approximated vector field (b).

the approximated vector field has visually the same global character as the original vector field.

5.1. Comparison with existing approach

The proposed approach for vector field approximation using RBF needs to be compared to a different competing approach. We selected an approach which uses the discrete Fourier transform (DFT) to approximate a vector field and a second approach which uses the discrete Cosine transform (DCT) to approximate a vector field.

The discrete Fourier transform computes for both components v_x and v_y of the

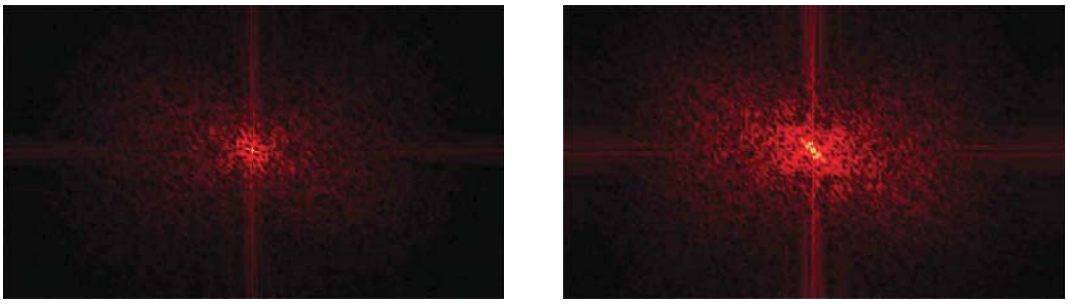
https://doi.org/10.1016/j.advengsoft.2018.06.013

vector field the Fourier transform using the formula

$$F(\alpha,\beta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}v(x,y)e^{-2\pi i(x\alpha+y\beta)}dxdy, \tag{30}$$

where α and β are frequencies that represent the original data set after the Fourier transform, see Figure 13. The inverse Fourier transform is computed using the formula

$$v(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F(\alpha,\beta)e^{2\pi i(x\alpha+y\beta)}d\alpha d\beta. \tag{31}$$



(a) Frequency portrait. (b) Phase offset.

Figure 13: Fourier transform of the original data set. Figure (a) represents the frequency portrait and (b) represents the phase offset (both images are rotated 90 counterclockwise).

The vector field can be approximated using the Fourier transform. We used the discrete Fourier transform in our experiments. Then we used only some number of low frequency coefficients, thereby removing the high frequencies from the input data set and simplifying the vector field. This simplified vector field can be compared with our proposed approach for vector field RBF approximation.

The discrete Cosine transform computes for both components v_x and v_y of the vector field the Cosine transform using the formula

$$F(\alpha,\beta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}v(x,y)\cos\left[\pi\left(x+\frac{1}{2}\right)\alpha\right]\cos\left[\pi\left(y+\frac{1}{2}\right)\beta\right]dxdy, \tag{32}$$

where α and β are frequencies that represent the original data set after the Cosine transform, see Figure 14. The inverse Cosine transform is computed using the formula

$$v(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F(x,y)\cos\left[\pi\left(x+\frac{1}{2}\right)\alpha\right]\cos\left[\pi\left(y+\frac{1}{2}\right)\beta\right]d\alpha d\beta. \tag{33}$$

https://doi.org/10.1016/j.advengsoft.2018.06.013

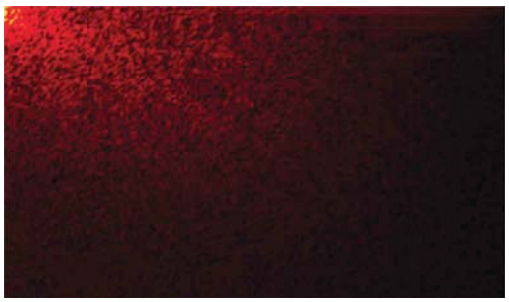


Figure 14: Frequency portrait of the Cosine transform of the original data set.

The vector field can be approximated using the Cosine transform. We used the discrete Cosine transform in our experiments. Then we used only some number of low frequency coefficients, thereby removing the high frequencies from the input data set and simplifying the vector field. This simplified vector field can be compared with our proposed approach for vector field RBF approximation.

The approximation error can be measured using different formulas. The first way is to compute the average difference of the approximated vector field and the original vector field. The average difference is computed using

$$Err = \frac{\sum_{i=1}^{N} \|\boldsymbol{v}_i - \bar{\boldsymbol{v}}_i\|}{N}, \tag{34}$$

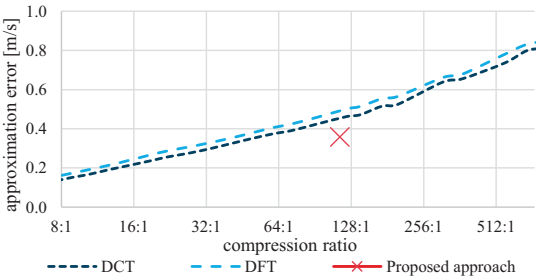
where $\boldsymbol{v}_i$ is the approximated vector, $\bar{\boldsymbol{v}}_i$ is the original vector and N is the number of the original samples. The approximation error for different compression ratios is visualized in Figure 15a. The total number of input points for vector field approximation is around $2.2 \cdot 10^4$; the average vector length, i.e. average speed, is $2.21\ m/s$.

Next, we can measure the average vector length error, i.e. the average speed error. This error is computed using

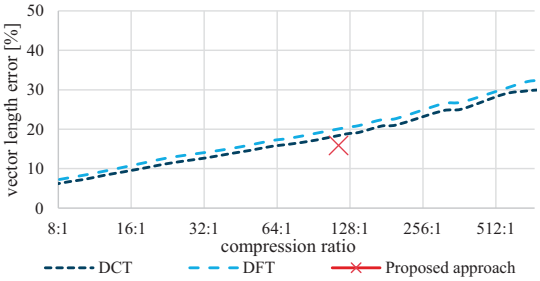
$$Err = \frac{\sum_{i=1}^{N} |\|\boldsymbol{v}_i\| - \|\bar{\boldsymbol{v}}_i\||}{\sum_{i=1}^{N} \|\bar{\boldsymbol{v}}_i\|}. \tag{35}$$

The average vector length error is divided by the average vector length of the original data set, i.e. this error is relative and the result for different compression ratios is visualized in Figure 15b. As the data set contains critical points (zero points), standard relative error computation using the following formula cannot be applied,

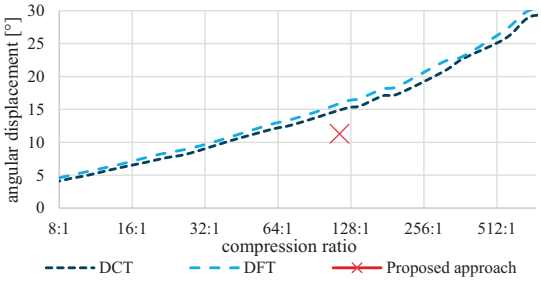
https://doi.org/10.1016/j.advengsoft.2018.06.013



(a) The average difference approximation error.



(b) The average vector length error.



(c) The average angular displacement error.

Figure 15: Comparison of approximation error for the proposed RBF vector field approximation with the DFT vector field approximation and the DCT vector field approximation for different compression ratios. The total number of input points for vector field approximation is around $2.2 \cdot 10^4$; for approximation with compression ratio $8:1$ are used $2.8 \cdot 10^3$ reference points; and for compression ratio $512:1$ *only* 43 *reference points* are used.

as there will be a division by zero and thus infinitely large relative error.

$$Err = \frac{1}{N} \sum_{i=1}^{N} \frac{|\|\boldsymbol{v}_i\| - \|\bar{\boldsymbol{v}}_i\||}{\|\bar{\boldsymbol{v}}_i\|}. \tag{36}$$

Last, we can measure the average angular displacement error using

$$Err = \frac{1}{N} \sum_{i=1}^{N} acos\left(\frac{\boldsymbol{v}_i \cdot \bar{\boldsymbol{v}}_i}{\|\boldsymbol{v}\| \, \|\bar{\boldsymbol{v}}\|}\right). \tag{37}$$

It can be seen that our proposed method have lower approximation error in all the three cases mentioned before.

As the next step we tested the number of critical points in the approximated vector fields. Our aim is to reduce the number of critical points and preserve only

https://doi.org/10.1016/j.advengsoft.2018.06.013

the important critical points. However, if we are reducing the critical points even more we are still able to preserve more critical points in the approximated vector field than the DFT and the DCT approximation, see Figure 16.

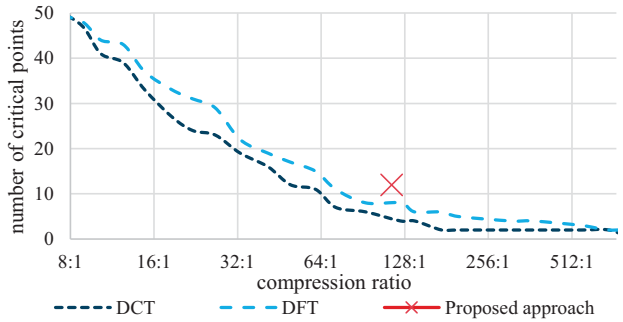


Figure 16: Comparison of the number of critical points for the proposed RBF vector field approximation and the DFT and the DCT vector field approximation for different compression ratios. The total number of critical points in the input data set is 69.

In spite of the fact that we know the distribution of approximation errors, it is hard to compare the two histograms of approximation errors in Figure 17. Therefore we computed the difference histogram, see Figure 18.

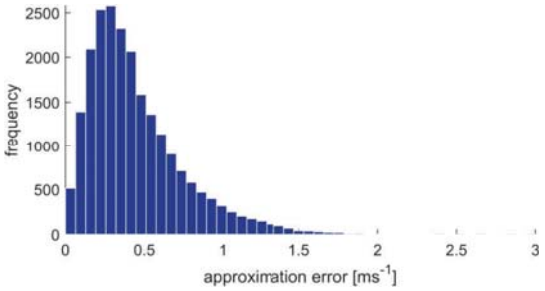
It can be seen that the proposed RBF approximation method has more smaller approximation errors and fewer higher approximation errors than the discrete Fourier approximation as well as the discrete cosine transform approximation, which means that the proposed method is significantly more accurate while preserving the same compression ratio and important features of the vector field.

6. Conclusion

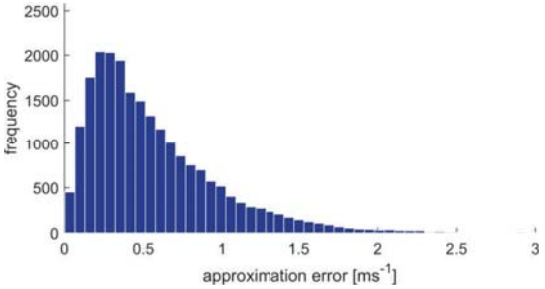
We presented a new approach for simplification and approximation of a vector field using radial basis functions. Important critical points are preserved in the approximated vector field. The algorithm proved its simplicity and ability to approximate a complex vector field. The proposed algorithm was compared with the standard Fourier approximation algorithm and the Cosine approximation algorithm. It proved its capability of high compression while maintaining a low approximation error. The proposed method also leads to an analytical RBF form of the vector field which can be used for further processing. This is a significant advantage over other methods.

In the future the proposed approach will be extended to approximate $3D$ vector fields, as the visualization of complex and noisy vector fields can be confusing.

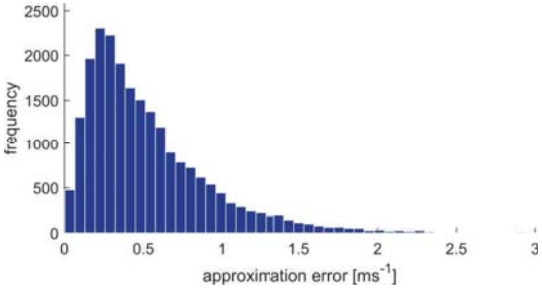
https://doi.org/10.1016/j.advengsoft.2018.06.013



(a) Proposed RBF approximation.

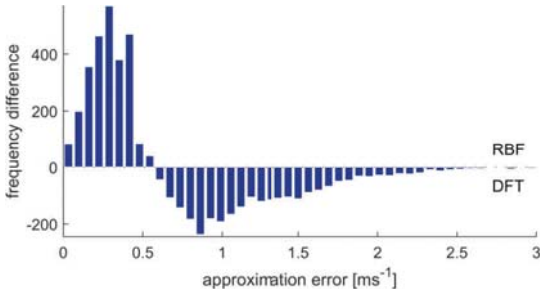


(b) DFT approximation.

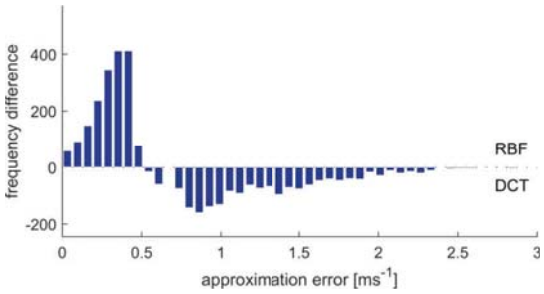


(c) DCT approximation.

Figure 17: The histograms of vector field average difference approximation error for compression ratio 115 : 1.



(a)



(b)

Figure 18: The difference histogram of vector field average difference approximation error ("RBF" - "DFT") (a) and ("RBF" - "DCT") (b), see Figure 17, for compression ratio 115 : 1.

Acknowledgment

The authors would like to thank their colleagues at the University of West Bohemia, Plzen and colleagues at the Institute of computer science, the Czech academy

https://doi.org/10.1016/j.advengsoft.2018.06.013

of sciences, for their discussions and suggestions, and anonymous reviewers for their valuable comments and hints provided. The research was supported by project Czech Science Foundation (GACR) No. 17-05534S and partially by SGS 2016-013.

References

[1] P. J. Davis, Interpolation and approximation, Courier Corporation, 1975.

[2] J. O'Rourke, A. J. Mallinckrodt, et al., Computational geometry in C, Computers in Physics 9 (1) (1995) 55–55.

[3] R. S. Laramee, H. Hauser, L. Zhao, F. H. Post, Topology-based flow visualization, the state of the art, in: Topology-based methods in visualization, Springer, 2007, pp. 1–19.

[4] X. Tricoche, G. Scheuermann, H. Hagen, Continuous topology simplification of planar vector fields, in: Proceedings of the conference on Visualization'01, IEEE Computer Society, 2001, pp. 159–166.

[5] X. Tricoche, G. Scheuermann, H. Hagen, A topology simplification method for 2D vector fields, in: Visualization 2000. Proceedings, IEEE, 2000, pp. 359–366.

[6] T. Weinkauf, H. Theisel, K. Shi, H.-C. Hege, H.-P. Seidel, Extracting higher order critical points and topological simplification of 3D vector fields, in: Visualization, 2005. VIS 05. IEEE, IEEE, 2005, pp. 559–566.

[7] H. Theisel, C. Rossl, H.-P. Seidel, Combining topological simplification and topology preserving compression for 2D vector fields, in: Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on, IEEE, 2003, pp. 419–423.

[8] T. K. Dey, J. A. Levine, R. Wenger, A delaunay simplification algorithm for vector fields, in: Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on, IEEE, 2007, pp. 281–290.

[9] P. C. Wong, H. Foote, R. Leung, E. Jurrus, D. Adams, J. Thomas, Vector fields simplification-a case study of visualizing climate modeling and simulation data sets, in: Visualization 2000. Proceedings, IEEE, 2000, pp. 485–488.

[10] W. de Leeuw, R. van Liere, Multi-level topology for flow visualization, Computers & Graphics 24 (3) (2000) 325–331.

https://doi.org/10.1016/j.advengsoft.2018.06.013

[11] P. Skraba, B. Wang, G. Chen, P. Rosen, 2D vector field simplification based on robustness, in: Visualization Symposium (PacificVis), 2014 IEEE Pacific, IEEE, 2014, pp. 49–56.

[12] P. Skraba, B. Wang, G. Chen, P. Rosen, Robustness-based simplification of 2D steady and unsteady vector fields, IEEE transactions on visualization and computer graphics 21 (8) (2015) 930–944.

[13] D. Günther, A. Jacobson, J. Reininghaus, H.-P. Seidel, O. Sorkine-Hornung, T. Weinkauf, Fast and memory-efficienty topological denoising of 2D and 3D scalar fields, IEEE transactions on visualization and computer graphics 20 (12) (2014) 2585–2594.

[14] D. A. C. Cabrera, P. Gonzalez-Casanova, C. Gout, L. H. Juárez, L. R. Reséndiz, Vector field approximation using radial basis functions, Journal of Computational and Applied Mathematics 240 (2013) 163–173.

[15] S. Koch, J. Kasten, A. Wiebel, G. Scheuermann, M. Hlawitschka, 2D vector field approximation using linear neighborhoods, The Visual Computer 32 (12) (2016) 1563–1578.

[16] J. Blazek, Computational fluid dynamics: principles and applications, Butterworth-Heinemann, 2015.

[17] B. Sanderse, S. Pijl, B. Koren, Review of computational fluid dynamics for wind turbine wake aerodynamics, Wind energy 14 (7) (2011) 799–819.

[18] F. Molina-Aiz, H. Fatnassi, T. Boulard, J. Roy, D. Valera, Comparison of finite element and finite volume methods for simulation of natural ventilation in greenhouses, Computers and electronics in agriculture 72 (2) (2010) 69–86.

[19] E. T. Chung, Y. Efendiev, C. S. Lee, Mixed generalized multiscale finite element methods and applications, Multiscale Modeling & Simulation 13 (1) (2015) 338–366.

[20] J. Helman, L. Hesselink, Representation and display of vector field topology in fluid flow data sets, IEEE Computer 22 (8) (1989) 27–36.

[21] R. Pan, V. Skala, A two-level approach to implicit surface modeling with compactly supported radial basis functions, Engineering with Computers 27 (3) (2011) 299–307.

https://doi.org/10.1016/j.advengsoft.2018.06.013

[22] G. E. Fasshauer, Meshfree approximation methods with MATLAB, Vol. 6, World Scientific, 2007.

[23] V. Skala, Meshless interpolations for computer graphics, visualization and games, in: Eurographics 2015 - Tutorials, 2015.

[24] E. Larsson, B. Fornberg, A numerical study of some radial basis function based solution methods for elliptic PDEs, Computers & Mathematics with Applications 46 (5) (2003) 891–902.

[25] X. Zhang, K. Z. Song, M. W. Lu, X. Liu, Meshless methods based on collocation with radial basis functions, Computational mechanics 26 (4) (2000) 333–343.

[26] K. Uhlir, V. Skala, Reconstruction of damaged images using radial basis functions, in: Signal Processing Conference, 2005 13th European, IEEE, 2005, pp. 1–4.

[27] A. Karim, H. Adeli, Radial basis function neural network for work zone capacity and queue estimation, Journal of Transportation Engineering 129 (5) (2003) 494–503.

[28] S. Ghosh-Dastidar, H. Adeli, N. Dadmehr, Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection, IEEE Transactions on Biomedical Engineering 55 (2) (2008) 512–518.

[29] L. Yingwei, N. Sundararajan, P. Saratchandran, Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm, IEEE Transactions on neural networks 9 (2) (1998) 308–318.

[30] Z. Majdisova, V. Skala, Big geo data surface approximation using radial basis functions: A comparative study, Computers & Geosciences 109 (2017) 51–58.

[31] R. Pan, V. Skala, Surface reconstruction with higher-order smoothness, The Visual Computer 28 (2) (2012) 155–162.

[32] G. Prakash, M. Kulkarni, U. Sripati, Using RBF neural networks and Kullback-Leibler distance to classify channel models in free space optics, in: Optical Engineering (ICOE), 2012 International Conference on, IEEE, 2012, pp. 1–6.

[33] I. P. Schagen, Interpolation in two dimensions - a new technique, IMA Journal of Applied Mathematics 23 (1) (1979) 53–59.

https://doi.org/10.1016/j.advengsoft.2018.06.013

[34] Z. Majdisova, V. Skala, Radial basis function approximations: Comparison and applications, Applied Mathematical Modelling 51 (2017) 728–743.

[35] V. Skala, RBF interpolation with CSRBF of large data sets, Procedia Computer Science 108 (2017) 2433–2437.

[36] H. Wendland, Computational aspects of radial basis function approximation, Studies in Computational Mathematics 12 (2006) 231–256.

[37] R. L. Hardy, Multiquadric equations of topography and other irregular surfaces, Journal of geophysical research 76 (8) (1971) 1905–1915.

[38] V. Skala, Fast interpolation and approximation of scattered multidimensional and dynamic data using radial basis functions, WSEAS Transaction on Mathematics 12 (5) (2013) 501–511.

[39] J. Lee, M. C. Strazicich, Minimum lagrange multiplier unit root test with two structural breaks, The Review of Economics and Statistics 85 (4) (2003) 1082–1089.

[40] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, A distributed lagrange multiplier/fictitious domain method for particulate flows, International Journal of Multiphase Flow 25 (5) (1999) 755–794.

[41] I.-S. Liu, Method of lagrange multipliers for exploitation of the entropy principle, Archive for Rational Mechanics and Analysis 46 (2) (1972) 131–148.

[42] J. Reininghaus, N. Kotava, D. Guenther, J. Kasten, H. Hagen, I. Hotz, A scale space based persistence measure for critical points in 2D scalar fields, IEEE Transactions on Visualization and Computer Graphics 17 (12) (2011) 2045–2052.

[43] W. Wang, W. Wang, S. Li, Detection and classification of critical points in piecewise linear vector fields, Journal of Visualization 1–15.

[44] S. Mann, A. Rockwood, Computing singularities of 3D vector fields with geometric algebra, in: Proceedings of the conference on Visualization'02, IEEE Computer Society, 2002, pp. 283–290.

[45] H. Bhatia, A. Gyulassy, H. Wang, P.-T. Bremer, V. Pascucci, Robust detection of singularities in vector fields, in: Topological Methods in Data Analysis and Visualization III, Springer, 2014, pp. 3–18.

https://doi.org/10.1016/j.advengsoft.2018.06.013

[46] V. Skala, Total least square error computation in E2: A new simple, fast and robust algorithm, in: Proceedings of the 33rd Computer Graphics International, ACM, 2016, pp. 1–4.

[47] L. Corbet, O. Vlček, K. Eben, J. Liczki, N. Benešová, M. Modlík, Regional air quality forecasting for the Czech Republic, in: 15th International Conference on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes, HARMO 2013, 2013, pp. 150–154.

https://doi.org/10.1016/j.advengsoft.2018.06.013