Automatic flow visualization method using self-organizing map

Masato MASUDA* and Yoshiaki TAMURA**
* Agricultural and Life Sciences, The University of Tokyo
1-1-1 Yayoi, Bunkyo-ku, Tokyo, 113-8657, Japan
E-mail: masuda.masato@mail.u-tokyo.ac.jp
** Information Sciences and Arts / Center for Computational Mechanics Research, Toyo University
2100 Kujirai, Kawagoe, Saitama, 350-8585, Japan

Received: 27 September 2021; Revised: 11 November 2021; Accepted: 13 December 2021

Abstract
In this paper, we propose a new visualization method using self-organizing map (SOM) for computed results of fluid flow. Most of existing visualization method gives color depending on a certain physical value, such as pressure, vorticity, etc. However the choice of the physical value is arbitrary and sometimes loses important features. The present method firstly classifies all flow properties, i.e. pressure, velocity components and their spatial gradients at each grid point, by giving these properties as high order vectors to SOM. Then color is given to each grid point based on its location on the map so that the flow field be naturally painted including all flow properties. SOM is originally a two-dimensional (2D) map as the main purpose of SOM is to visualize high order vectors, but in the present study the map is not directly viewed and only used to determine color of each grid point. Hence we try to generate three-dimensional (3D) SOM to give three color components based on the map. Both 2D and 3D map results are shown to demonstrate the capability of the present method.

Keywords : Visualization, Computational Fluid Dynamics(CFD), Self-Organizing Map(SOM), Color contour, Feature tracking, Machine learning

1. Introduction

Visualization of computed results of fluid flow has long history of research. The visualization methods are categorized into two groups; one is to visualize the physical values directly obtained from the computed results such as pressure, vorticity, etc. and the other is to visualize flow features such as vortices, separation regions, shock waves, etc. The former utilizes color contours or iso-surfaces for scalar functions and arrows for vector functions (for example, Buning and Steger, 1985). The latter is called feature tracking, that is a method to detect center of vortices or other flow features and visualize them using markers or surfaces (for example, Sawada, 1995). In all cases, researchers have definite purpose and choose flow functions. These methods do work when what to observe is clear. On the other hand, important features may be dropped off by fixing the flow functions when the computed flow fields are not well understood in advance.

Flow visualization methods mentioned above are computationally or mathematically recognized as methods reducing orders of high dimensional vector of each grid point to low (one) dimensional vector and give color based on the value. The mechanism to reduce order is prescribed in all the method. Hence we can make a new visualization method when we find a new formula to reduce order.

The self-organizing map (SOM) developed by Kohonen (1995) is a sort of machine learning for reducing dimensions of input high order vectors to two-dimensions and making a two-dimensional (2D) map to categorize the input vectors.

As described above, flow visualization consists of two steps. The first step is to choose a scalar function, in other words, to reduce dimensions of high order vector to one. Then color is given to each grid point in the next step. In the present research, the first step is replaced with SOM, which automatically reduce dimensions. When we give all flow properties, e.g. pressure, velocity, etc. and their spatial gradients to SOM, SOM may categorize the flow without dropping any flow features off.
In the literature, machine learning is being used for fluid analysis. For example, Kutz models turbulence with machine learning (Kutz, 2017). Stoecklein et al. use machine learning for flow control (Stoecklein et al., 2017) and Matsuoka et al. predict flow from the visualized images by machine learning (Matsuoka et al., 2018). However any research using machine learning for visualization has not been conducted as far as the authors know.

In the next section, SOM used in this research is described. The following sections includes overview of sample flow fields and their visualized results, along with the discussion. The conclusions are mentioned in the final section.

2. Classification of flow field by SOM

SOM is a two layer neural network with an input layer and a competitive layer (map). Multiple input high order vectors are displayed on the map illustrating the relationship among the input vectors by clustering them with their similarity. Moreover, the resultant map has capability of interpolation and can put an unknown input vector at the proper location on the map. The procedure of the present research is described below. Two-dimensional incompressible flow fields are used as examples. Pressure, velocity vector components of \( x, y \) directions, e.g. \((u, v)\) at each grid point together with the velocity magnitude, \( x, y \) spatial derivatives (spatial gradients) of the pressure and \( u \) and \( v \), in total tenth order vector, are selected as the input vector components. The spatial gradients may imply the vorticity. The second derivatives of the velocity components are another candidates, especially for viscosity dominant flow, but are not selected in order to suppress the increase of the dimensions of the input vector. This effect will be discussed later.

SOM is generally a two-dimensional map, consisting of squares or hexagons, to visualize the classified input vectors. In this research, however, we do not directly view the map but only use for the classification. As color has three components (RGB, or others), three-dimensional SOM is possible for giving one color to one grid point. Thus \( n \times n \) square and \( n \times n \times n \) cubic SOM are generated here. The initial values of tenth order vectors are randomly generated and given to each cell of the map and learned with the input vectors. The neighborhood radius and the learning coefficient monotonically decrease as \( n/2 \rightarrow 1 \) and \( 0.3 \rightarrow 0.01 \) with the epoch (step) of the learning, followed by Masuda, et al. (2012). The boundaries of the map can be periodic. Periodic or non-periodic boundaries are selected based on the periodicity of the color component. When the color component is hue, the boundary should be periodic. Otherwise the boundary can be non-periodic. The training data are picked up from the computed results considering the variety of the vectors. This will be described later in detail.

After learning, input of the tenth order vector at a grid point ignites a certain cell of the map. The location of the cell indicates the feature of the vector. The ignited cell has two coordinate values on the 2D map and three coordinate values on the 3D map. These values range from 1 to \( n \), according to the map size. Color contour is made by assigning a color corresponding to the feature values. The choice of color components is primarily RGB, but HSV (hue, saturation, value) is another candidate. These are tested by trial and error manner.

3. Computation and visualization using SOM

3.1. Overview of the first example

The first example is two-dimensional flow around a circular cylinder. In order to detect the wake region, Cartesian grid of 1280×800 is used with the cylinder diameter of 40 cells, resulting 32×20 computational area normalized by the diameter. The Reynolds number is set 10,000 in order to observe continuous vortex shedding. The cell width is 1/40 of the diameter. The computational area is not sufficiently large and the cell width is not sufficiently small but enough to discuss visualized images. The solution algorithm is pseudo-compressibility method with inner iteration for unsteady flow. The time step is 6.25×10\(^{-3}\) non-dimensionlized second normalized by the diameter of the cylinder and the uniform flow velocity. The advection terms are discretized with third-order upwind scheme (MUSCL) and the viscous terms are discretized with second-order central difference. The time integration is Lower-Upper Symmetric Gauss-Seidel (LU-SGS). For the detail of the numerical method, please refer to CFD text books (for example, Hirsh, 2007, Anderson et al., 2020). Figure 1 shows an instantaneous pressure and vorticity contours. Obviously flow is disturbed only around the cylinder and the wake. Thus the input vectors are selected within a black line box in Fig. 1. The number of grid points in the box is 25,000 which is too much for the learning. Thus the following procedure reduces the number of input vectors. All of the vectors in the box is sorted by one of ten vector components and take 100 vectors at constant intervals. The same procedure is done for the other nine components. Then we have almost 1000 candidates of the input vector, which may cover most of the flow features in the box even though the number of input vectors reduces to 1/25.
3.2. Generation of SOM

Several combination of size and dimension of SOM are examined and summarized in Table 1. In each epoch,

| map size | no. of epochs | boundary condition |
|----------|---------------|--------------------|
| 20×20    | 1000          | periodic / periodic |
| 128×128  | 4000          | periodic / non-periodic |
| 25×25×25 | 2000          | periodic / periodic / periodic |
| 35×35×35 | 3500          | periodic / periodic / periodic |

determining the neighborhood radius and the learning coefficient, searching the cell whose (Euclid) distance from each input vector is the closest, the vectors of cells within the neighborhood radius are modified with the learning coefficient. This process is repeated the given number of epochs. The number of epochs is determined by the results of the distributions of vector components as observed in Fig. 2 and Fig. 3, shown later. The order of input vector to the map is random. The input vectors are normalized from 0 to 1 in each dimension. The learning time is simply proportional to the number of input vectors, the number of cells of the map and the number of epochs. For example, the fourth case of Table 1 takes 1 and half hours on a 2.5GHz dual core Intel Core i7 PC. Distributions of each vector component on the map of 128² and 35³ are shown in Fig. 2 and in Fig. 3, respectively. Figure 2 shows color contours and Fig. 3 shows iso-surfaces. Even though the minimum and the maximum values are not necessarily 0 and 1 in all the components, the input vectors are

![Fig. 1 Computed result of flow around a circular cylinder at a certain time step. left: vorticity, right: pressure. Black line boxes show the area for SOM training.](image)
well classified.

Fig. 3 Distributions of each input vector components on SOM of $35^3$ with the ranges of their values.

3.3. Visualization using SOM

When visualizing the computed results, choosing the input vector of one grid point from the result, giving it to the trained map, a certain cell of the map ignites. The ignited cell has the vector closest to the input vector. Then the color corresponding to the coordinates of the ignited cell, such as $(x_m, y_m)$ ($0 \leq x_m, y_m \leq 1$) for 2D map and $(x_m, y_m, z_m)$ ($0 \leq x_m, y_m, z_m \leq 1$) for 3D map, is considered to be the color of the grid point. Finally the collection of colors of all of the grid points makes an image of the color contour.

Figure 4 illustrates this procedure schematically.

(1) choose one high order vector from the result (a) and give to the trained SOM (b) to find out the cell to ignite. Note that even the SOM is generated with a limited number of input vectors, any grid point either inside or outside of the box in Fig. 1 must have one ignited cell in the map.

(2) pick up the color from the color map (c) corresponding the cell of the trained SOM (b).

(3) give the color to the location of the chosen vector to make a color contour.

Fig. 4 Schematic picture of procedure to visualize a computed result using a trained SOM.
For the correspondence between the map coordinates and the color, the following conversion functions are adopted for 2D map.

\[
R = 2|x_m - 0.5|, \quad G = 2 \times |y_m - 0.5|, \quad B = 0 \tag{1}
\]

\[
H = \text{frac}(x_m + H_{\text{off}}), \quad S = 1, \quad V = y_m \tag{2}
\]

Note that \((H, S, V)\) is an HSV color model. \(\text{frac}(\cdot)\) is a function to extract a fractional part. \(H_{\text{off}}\) is an offset of hue. For 3D map,

\[
[R \text{ or } G \text{ or } B] = \frac{1}{2} \left[ \sin(2\pi[x_m \text{ or } y_m \text{ or } z_m] + [R \text{ or } G \text{ or } B]_{\text{off}}) + 1 \right] \tag{3}
\]

The correspondence between \((R, G, B)\) and \((x_m, y_m, z_m)\) (for example, \(R \leftrightarrow x_m\) or \(R \leftrightarrow y_m\) or \(R \leftrightarrow z_m\)) is determined simply by the impression of the images.

There are a plenty of combinations of the map size, the map dimensions and the conversion functions. Only a couple of them are presented here. Firstly Fig. 5 shows the 2D map results. The left is the result of the 20×20 map with Eq. (1) and the right the 128×128 map with Eq. (2) where \(H_{\text{off}} = 0.2\).

![Fig. 5 2D map results. Left; 20×20 map with Eq. (1). Right; 128×128 map with Eq. (2).](image)

![Fig. 6 2D colored maps used for Fig. 5. Left; Eq. (1) (20×20). Right; Eq. (2) (128×128).](image)

The both pictures clearly show the flow features of the pressure contour and vorticity contour in Fig. 1. Especially the wake region is clearer than in Fig. 1. Even though the input vectors are selected within a small region (the black line box in Fig. 1), the region outside of the box is also well classified. Comparing the two picture, the larger the map size is, the color gradation is smoother simply because the number of colors used is equal to the number of the cells of the map. Figure 6 visualizes Eq. (1) and Eq. (2) used for Fig. 5.

| map size | \(R\) | \(R_{\text{off}}\) | \(G\) | \(G_{\text{off}}\) | \(B\) | \(B_{\text{off}}\) |
|----------|------|------|------|------|------|------|
| 35×35   | 0.4\(\pi\) | 0.7\(\pi\) | 0.5\(\pi\) | 0.4\(\pi\) | 0.7\(\pi\) | 0.5\(\pi\) |

The result with 35×35 3D map is shown in Fig. 7 and the parameters in Eq. (3) is tabulated in Table 2. Figure 7 shows the time sequence of every 0.625 non-dimensionalized second. Even though the map is only generated by the result of the first image, the second and subsequent time results are well classified and visualized. Figure 8 shows the colored 3D map. The parameters are those in Table 2.

![Table 2 Parameters for visualizations using 3D map in Eq. (3).](image)
3.4. Effect of input vector components

In order to examine the effect of the choice of input vector components, two kinds of input vectors are tried. One is input vectors of only \((u, v, p)\) and the other is input vectors of 6 spatial gradients, e. g. \((p_x, p_y, u_x, u_y, v_x, v_y)\). All of the parameters to generate maps are the same as the case of 2D map with the size of 128×128. Only the visualized flow field images are presented in Fig. 9. The left is generated with \((u, v, p)\) map, the middle is with 6 spatial gradients and the right is with full components (the same as Fig. 5 right). All three images show flow features. However the left is rather noisy compared with the other two and the middle misses some details, such as blue-green region around the center of the right picture, which is distinguished from the other region, but not in the middle picture. In this particular case, the full component vector gives the best result.

3.5. Another example

In order to confirm applicability of the present method to other types of flow field, a two-dimensional driven cavity
flow is examined. The reason why we choose a driven cavity flow is that
- driven cavity is steady while flow around a cylinder is unsteady,
- driven cavity is shear flow dominant while flow around a cylinder is advection dominant.

As the second derivatives are not selected as input vector components in this research, a driven cavity is suitable for examining viscosity effect. The computational area is square divided by 1000x1000. The Reynolds number is set to be 5,000. The numerical method is the same as the previous case.

Figure 10 shows the result with standard visualization, namely, pressure, velocity components and vorticity contours and streamlines. The accuracy of the result is confirmed by comparing with a previous work (Erturk, et al., 2005).

![Fig. 10 Pressure, u, v, vorticity and streamlines of the driven cavity.](image)

This result is again visualized with 2-D and 3-D map. Input vectors for training are selected from the region near the boundaries. More exactly the grid points whose distance from the boundary is less than 0.1 (the cavity size is 1x1) is adopted because circulated flows occur and the physical values change near the boundaries rather than the middle of the cavity. Again the selected input vectors are sorted and picked up with respect to each vector component, resulting in about 650 vectors. The parameters are summarized in Table 3.

| map size  | no. of epochs | boundary condition      |
|-----------|---------------|-------------------------|
| 128x128   | 4000          | non-periodic / non-periodic |
| 35x35x35  | 4000          | periodic / periodic / periodic |

Figure 11 shows the visualized images by 2-D map (left) and 3-D map(right). For 2-D map, Eq. (2) is used and for 3-D map, Eq. (3) is used with the parameters in Table 4. Both 2-D and 3-D maps visualize the flow features observed in Fig. 10. Similar to the result of flow around a cylinder, 3-D map depicts the details of the flow.

![Table 4 Parameters for visualization of driven cavity using 3D map in Eq. (3).](image)

4. Discussions

From the visualizations of flow around a circular cylinder, the present proposed method of SOM visualization practically works well. As all of the physical values are given to SOM, the resultant images do not depict information about each physical values, namely, the highest/lowest pressure regions or strong vortex regions, and so on. However the entire flow pattern is clearly illustrated, especially the shedding vortices and the wake region. For the driven cavity, the same procedure successfully classifies the flow inside of the cavity where the shear (viscous) flow is dominant.

In the present paper, the components of input vector were basically fixed. Only 2D maps of 128x128 size were generated with the three types of components. Also as mentioned in 2., second derivatives are not included. Even though the two flow fields were successfully visualized, the choice of components will be further examined.

About the map size, the larger is the better, in these two example cases. As the map size directly affect the computational time, optimal size is to be pursued but is not still determined with only two cases. The conversion function from \((x_m, y_m, z_m)\) to \((R, G, B)\) or \((H, S, V)\) with a couple of parameters is also an open question. Here we chose them from the visualized image by trial and error manner. Only small change of the offset value in Eq. (3) gives an output of different impression.
5. Conclusions

A new visualization method using SOM was proposed. Physical values (and their spatial gradients) of each grid point are given to SOM as high order input vectors and classified automatically. From the classified result of the map, colors are assigned to each cell of the map and utilized to draw color contours. The map is two-dimensional or three-dimensional. Both maps are examined with changing map size. Within the present limited cases, larger map (larger number of cells) gives clearer representation of flow features.

For future works, another types of flow fields are to be investigated, such as compressible flows and multiphase flows as well as three-dimensional flows.

The other works are interpretation of the image generated by the present method and consideration of the practical use. The generated images actually give new impression of flows, but physical interpretation is still a disadvantage of the method. The present authors hope that numbers of trials of visualization by the present method give a new insight of flow field.

References

Anderson, D. A., et al., Computational Fluid Mechanics and Heat Transfer, 4th edition, CRC Press (2020).
Buning, P. G. and Steger, J. L., Graphics and visualization in computational fluid dynamics, AIAA Paper 85-1507 (1985).
Erturk, E., Corke, T. C. and Gökçöl, C., Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers, International Journal for Numerical Methods in Fluids Vol.48, No.7 (2005), pp. 747 - 774.
Hirsh, C., Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics, Second Edition, Butterworth-Heinemann (2007).
Kohonen, T., Self-organizing maps, Springer-Verlag, (1995).
Kutz, J. N., Deep learning in fluid dynamics, Journal of Fluid Mechanics, Vol.814 (2017), pp. 1-4.
Masuda, M., Nakabayashi, Y., and Yagawa, G., Radius parallel self-organizing map (RPSOM), Journal of Computational Science and Technology, Vol.6 No.1 (2012), pp.16–27.
Matsuoka, D., Nakano, M., Sugiyama, D., and Uchida, S., Deep learning approach for detecting tropical cyclones and their precursors in the simulation by a cloud-resolving global nonhydrostatic atmospheric model, Progress in Earth and Planetary Science, Vol.5, No.80 (2018).
Sawada, K., A Convenient Visualization Method for Identifying Vortex Centers, Transactions of the Japan Society for Aeronautical and Space Sciences, Vol.38, No.120 (1995), pp.102-116.
Stoecklein, D., Lore, K. G., Davies, M., Sarkar, S., and Ganapathysubramanian, B., Deep learning for flow sculpting: Insights into efficient learning using scientific simulation data, Scientific Reports Vol.7, 46368 (2017).