

A novel method of low-dimensional representation for temporal behavior of flow fields using deep autoencoder

Cite as: AIP Advances **9**, 015006 (2019); <https://doi.org/10.1063/1.5067313>

Submitted: 17 October 2018 . Accepted: 28 December 2018 . Published Online: 09 January 2019

Noriyasu Omata, and Susumu Shirayama 

COLLECTIONS

Paper published as part of the special topic on **Chemical Physics, Energy, Fluids and Plasmas, Materials Science** and **Mathematical Physics**



View Online



Export Citation



CrossMark

ARTICLES YOU MAY BE INTERESTED IN

Machine learning methods for turbulence modeling in subsonic flows around airfoils
Physics of Fluids **31**, 015105 (2019); <https://doi.org/10.1063/1.5061693>

Fast flow field prediction over airfoils using deep learning approach
Physics of Fluids **31**, 057103 (2019); <https://doi.org/10.1063/1.5094943>

A deep learning enabler for nonintrusive reduced order modeling of fluid flows
Physics of Fluids **31**, 085101 (2019); <https://doi.org/10.1063/1.5113494>

AIP Advances

Nanoscience Collection

READ NOW!

A novel method of low-dimensional representation for temporal behavior of flow fields using deep autoencoder

Cite as: AIP Advances 9, 015006 (2019); doi: 10.1063/1.5067313

Submitted: 17 October 2018 • Accepted: 28 December 2018 •

Published Online: 9 January 2019



View Online



Export Citation



CrossMark

Noriyasu Omata^{a)} and Susumu Shirayama

AFFILIATIONS

Department of Systems Innovation, The University of Tokyo, 7-3-1 Hongo, Bunkyo-Ku, Tokyo 113-8656, Japan

^{a)}omata@naki.tu-tokyo.ac.jp

ABSTRACT

A method capable of comparing and analyzing the spatio-temporal structures of unsteady flow fields has not yet been established. Temporal analyses of unsteady flow fields are often done after the data of the fields are reduced to low-dimensional quantities such as forces acting on objects. Such an approach is disadvantageous as information about the flow field is lost. There are several data-driven low-dimensional representation methods that preserve the information of spatial structure; however, their use is limited due to their linearity. In this paper, we propose a method for analyzing the time series data of unsteady flow fields. We firstly propose a data-driven nonlinear low-dimensional representation method for unsteady flow fields that preserves its spatial structure; this method uses a convolutional autoencoder, which is a deep learning technique. In our proposed method, the spatio-temporal structure can be represented as a trajectory in a low-dimensional space using the visualization technique originally proposed for dynamic networks. We applied the proposed method to unsteady flows around a two-dimensional airfoil and demonstrated that it could briefly represents the changes in the spatial structure of the unsteady flow field over time. This method was demonstrated to also be able to visualize changes in the quasi-periodic state of the flow when the angle of attack of the airfoil was changed. Furthermore, we demonstrated that this method is able to compare flow fields that are constructed using different conditions such as different Reynolds numbers and angles of attack.

© 2019 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/1.5067313>

I. INTRODUCTION

Highly accurate computational fluid dynamics (CFD) calculations and high-resolution experimental observations have been used to clarify the characteristics of unsteady flows.^{1,2} Along with the development of these methods, a large amount of flow data has been acquired and accumulated. It is expected that the performance of fluid devices can be improved by considering unsteadiness using these data in the design and development, and research is being conducted.^{3–5} However, the findings obtained by such investigations are not organized like those for steady and averaged fields; one of the reasons is that it is difficult to compare the spatio-temporal data obtained for unsteady fields.

In general, a method for comparing several unsteady flow fields has not yet been established. Comparing the time series data of two or more fields is the most commonly used method by studies, and it is usually performed using quantities such as the component of a field's velocity at a specific position and the force acting on a body in the flow. However, because it is difficult to grasp the spatial relevance with such quantities, this method is often used in conjunction with visualizations of instantaneous fields;^{6,7} an example of this can be seen in Figure 1, which is a vector representation of several instantaneous velocity fields. Visualizations of isocontour plots and instantaneous streamlines may also be performed as well. These types of methods, however, struggle to capture temporal changes because a single image can provide information

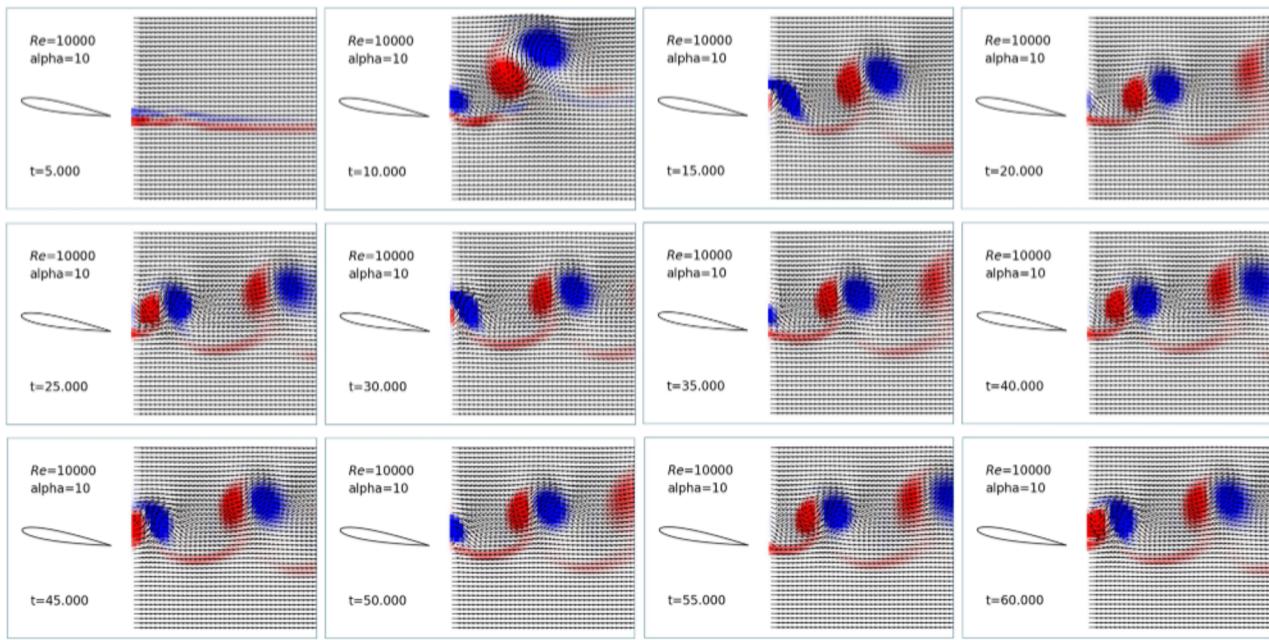


FIG. 1. Example of instantaneous flow fields. $Re = 10,000$, $\alpha = 10^\circ$.

only about either a specific time frame or over the entire time. In addition, because comparisons between the fields are often done by appearance, it is difficult to discern between changes that occur when the time steps are decreased; this makes detailed analyses more difficult to do. Animation, a method that continuously displays visualizations of instantaneous fields, is the most intuitive way to display temporal behavior. However, this method has a problem that a lot of information is presented in a short time. Therefore, it is difficult even for experts to compare multiple unsteady flows by animation. This is also problematic in terms of objectivity. In addition, the method requires an appropriate device, and it cannot be implemented on paper. How to visualize such spatio-temporal data, high dimensional big data more generally, is a common problem not only in fluid dynamics but also in many fields.

Conversely, various kinds of data were collected in many research fields. Methods to utilize these data is being sought and data-driven approach is drawing attention.⁸ This is no exception in the field of fluid dynamics. In fact, analysis of flow data by modal decomposition, for example, is performed as an analysis by a data-driven method.^{9,10} In addition, its application is expanding to vortices detection,¹¹ simulation,¹² classification,¹³ and prediction.¹⁴ Furthermore, in recent years, a method using a multi-layered neural network called deep learning has made remarkable achievements.^{15,16} Application of this method is also actively being made in fluid dynamics.¹⁷ For example, research on improvement of PIV,¹⁸ CFD,¹⁹ flow control²⁰ has been carried out. However, as far as the authors knowledge, deep learning techniques have not been used for flow data analysis.

In this paper, we propose a data-driven visualization method for comparing the spatio-temporal structure of unsteady flow fields. A novel low-dimensional feature representation of the spatial structure of the flow fields using deep learning is proposed and used in the visualization. The proposed method allows for a visualization of the unsteady flow fields in a feature space, and their temporal behavior of spatial structures can be visualized as trajectories in the feature space. We applied the method to unsteady flow around a two-dimensional airfoil. It is shown that temporal behaviors of the flow fields can be represented in the low-dimensional feature space. The low-dimensional representation makes it easy to compare the unsteady flow fields.

II. RELATED WORK

A. Low-dimensional expression of flow

To obtain information about temporal changes occurring to unsteady flows, studies have often mapped this type of flow field to some low-dimensional quantities; an example of this is time series data that were obtained at a specifically defined position, with information from all other points being discarded, and a further example is that of a time series of forces acting on a body such as lift coefficient shown in Figure 2. These quantities can be said to be an expression of an original flow field, and a comparison between them can be done in a low-dimensional feature space. Because these quantities are univariate, they are quantitatively comparable; for example, Fourier transforms, wavelet transformations,²¹ and the ARMA model²² are often applied to these quantities. However, when

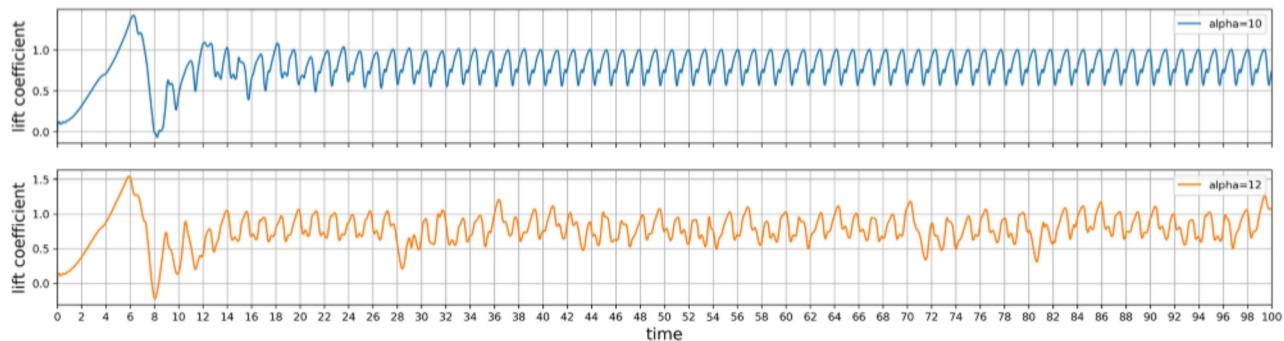


FIG. 2. Time-series graph of the lift coefficient. $Re = 10,000$, $\alpha = 10^\circ$ and 12° .

low-dimensionalized quantities such as these are used, large amounts of information related to the spatial structure of a flow field are lost.

One way in which this drawback can be resolved is to use data-driven approach. The most fundamental method for unsteady flow is a modal analysis that transform an entire flow field.^{9,10} In a study that carried out such an analysis, the proper orthogonal decomposition (POD)²³ and dynamic mode decomposition (DMD)²⁴ methods were used.

POD is equivalent to a principal component analysis (PCA)²⁵ technique. Each instantaneous flow field can be represented by a linear weighted sum of orthogonal basis vectors. The orthogonal basis vector is called as POD mode. A low-dimensional expression of the flow field can be derived from a few higher mode of POD. However, since the each instantaneous field is independently dealt with in POD, it is quite difficult to understand the temporal information of flow field by POD mode itself. In contrast, DMD has been developed to understand time evolving flow field. In DMD, one-time step change of unsteady flow field is expressed as a linear mapping such that $\mathbf{u}^{n+1} = \mathbf{A}\mathbf{u}^n$ where \mathbf{u} is a flow field, n denotes the time step and the matrix \mathbf{A} means a linear mapping operator. The DMD mode is derived from the eigenvectors of \mathbf{A} .

Research has been carried out to analyze flow fields with low-dimensional representations using these methods. However, because these transformation methods are linear and are based on strong assumptions, the type of flow fields that they can analyze properly is limited. In particular, Bright et al. pointed out that their POD-based methods can not cope with flow transitions, resulting in large reconstruction errors.²⁶

In this paper, deep learning,^{15,16} which is a state-of-the-art non-linear mapping method, is utilized as a more flexible way in which to express low-dimensional representations.

B. Visualization of spatio-temporal structures based on feature space

Understanding spatio-temporal data structures is difficult not only in the field of fluid dynamics but also in a variety

of academic fields. Van den Elzen et al. proposed a visualization method for dynamic (time-varying) networks that addresses this problem.²⁷ They focused on how to express the temporal information, which was difficult to obtain using conventional methods. In this subsection, we introduce this method, which is extended in our method.

In conventional visualizations of dynamic networks,²⁸ temporal changes are analyzed using visualizations of instantaneous networks that are displayed side-by-side.²⁹ In such methods, continuous changes cannot be determined, and as a result, it is difficult to understand what is happening as time progresses. Visualizations using animation are possible,³⁰ but it is difficult to analyze it quantitatively and to ensure objectivity. These difficulties in analyzing dynamic networks are considered to be similar to those faced in the analysis of unsteady flows.

To address this problem, van den Elzen et al. proposed a visualization method in which information about structural changes over time could be acquired. They proposed that an instantaneous network be represented by two-dimensional feature quantity expressed as a single point in a figure, and that the trajectory of the point in a display space represents structural changes of the network over time. This method proved to be advantageous, as it was able to show information about continuous temporal behavior. However, information related to a specific time is limited because it is represented by a single dot; furthermore, the structures of the networks cannot be determined without carrying out further visualizations.

The method consists of four steps: discretization, vectorization, dimension reduction, and visualization. The first step, discretization, acquires the state of an instantaneous network at a certain time. It can be said that this is a process of selecting data corresponding to a certain time from the data over the entire time period investigated. The next step, vectorization, is performed by arranging the values of each component of an adjacency matrix and some other features into a vector. The data at each time period are processed into a form that can be handled in the next step. In the third step, dimension reduction, the vector representation is transformed into

a two-dimensional feature quantity. Use of PCA²⁵ or t-SNE³¹ has been proposed in the literature. In the last step, visualization, the feature quantities of the network corresponding to each time period are plotted into a two-dimensional display space; the changes over time of the network are visualized as a trajectory that the point moved.

In this paper, we extend this method proposed so that it can be applied to flow fields, and we propose a method for obtaining data about structural changes over time.

III. PROPOSED METHOD

In this section, we propose a new method that builds a low-dimensional representation of the spatio-temporal structure of unsteady flows. To begin with, we describe the data that will be handled. Following this, we propose a method that represents the spatial structure of an instantaneous flow field as a low-dimensional quantity; the method uses a deep learning approach. Lastly, a method for visualizing temporal behavior of unsteady flow fields using the low-dimensional quantity is proposed.

A. Data used in the proposed method

In our method, flows with a variety of physical parameters such as the Reynolds number Re and the angle of attack α are compared, and let the set of these parameters be represented

by $\theta = (Re, \alpha)$. The set of θ to be calculated and be available is represented by Θ . Flows are intended to be obtained by CFD calculation, and an example of computational grid used is shown in gray in Figure 3.

Since we use the results of CFD simulations, the flow is temporally discretized. We represent dimensionless time with the character t and the time step of the calculations dt . Data sampling was performed with some time interval, and this interval is represented by Δt . The index of data sampling time is represented by n , and the number of samples is represented by N_t .

Additionally, to obtain data that are suitable for comparison, the data for each calculation have to be acquired at the same position in a sampling area in the different flow fields. Because we are comparing two-dimensional flow fields in particular in this paper, we use a rectangular area of $(x_s, y_s) - (x_e, y_e)$ as the sampling area as shown in red in Figure 3. Lattice-like sampling points of $N_x \times N_y$ are placed in the space, and two-dimensional velocity components, u and v , are extracted at each sampling point. The $N_x \times N_y \times 2$ dimensional data that are extracted represent each instantaneous field and are used as input.

Information about a flow field at the n -th sampling step corresponding to the flow θ are therefore represented by $N_x \times N_y \times 2$ -dimensional array, and let us denote it by \mathbf{X}_n^θ .

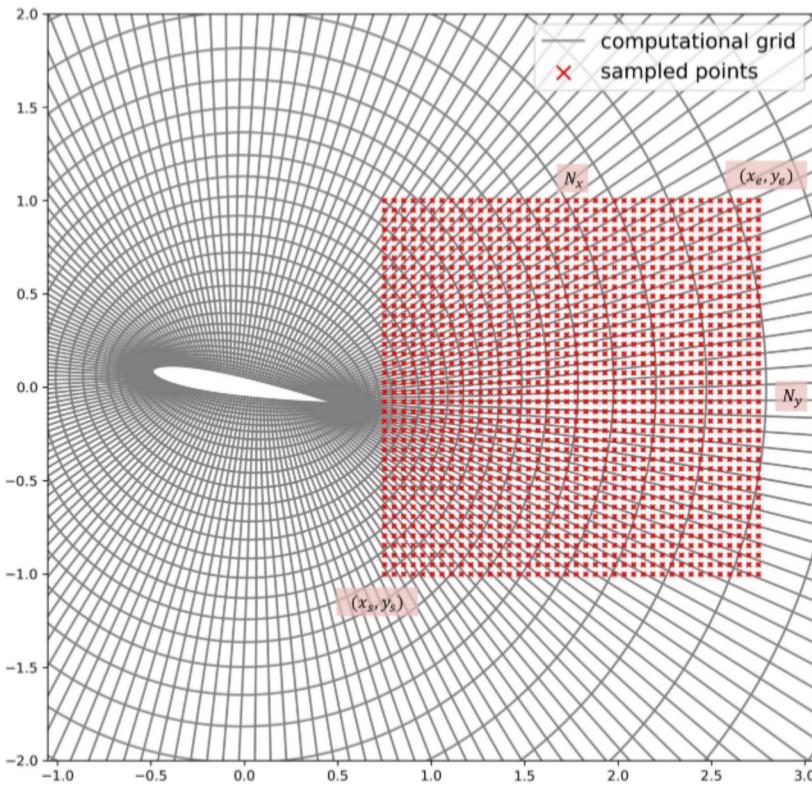


FIG. 3. Example of the computational grid and the sampling area.

B. Deep-learning-facilitated low-dimensional representation of flow fields

In our proposed method, the features of the spatial structures of instantaneous flow fields are extracted by low-dimensionalizing the multi-dimensional array data of the flow fields using a deep learning method called an autoencoder (AE),¹⁵ whose elements are convolution neural networks (CNNs).¹⁶

CNNs are neural networks that repeatedly apply local linear convolution operations and non-linear transformations. They can obtain global features by repeatedly obtaining data about local features using only a small number of parameters. Because of these characteristics, CNNs excel at dealing with high-dimensional information, especially in the field of image recognition.¹⁶ Because the data of flow fields is closely related to that of near region and can be obtained like images, CNNs can gradually determine information about the entirety of a feature solely from local information.

AE¹⁵ is a dimension reduction method, and it consists of an encoder function and a decoder function constructed with neural networks, as illustrated by the following equation:

$$\mathbf{Z} = \text{encoder}(\mathbf{X}), \quad \tilde{\mathbf{X}} = \text{decoder}(\mathbf{Z}). \quad (1)$$

In this equation, \mathbf{X} denotes original data, which in this paper is an instantaneous flow field, \mathbf{Z} denotes the feature quantities that represent the data, and $\tilde{\mathbf{X}}$ denotes the data reconstructed from a feature. Essentially, the encoder transforms \mathbf{X} to \mathbf{Z} , and the decoder projects the feature \mathbf{Z} to $\tilde{\mathbf{X}}$ in a data space. Each function is optimized as in the following equation:

$$\min_{\text{encoder,decoder}} D(\mathbf{X}, \tilde{\mathbf{X}}) \quad (2)$$

using the function D , which measures the difference between the original data \mathbf{X} and the reconstructed data $\tilde{\mathbf{X}}$. This optimization is done so that the result obtained by the decoding $\tilde{\mathbf{X}}$ become as close as possible to the original data \mathbf{X} . The dimension of the feature space, M , is generally set so as to be lower than that of the data space, N . Thus the feature, \mathbf{Z} , becomes an representation that holds the information necessary for restoring the data, despite its low dimensionality. We construct the encoder and the decoder functions using multi-layered CNNs.

An outline of the method to represent the spatial structure of flow fields by low-dimensional features using CNN-based AE (CNN-AE) is shown in Figure 4(a). The data input into the encoder and the data output by the decoder in the CNN-AE are information about an instantaneous flow field at a specific time, \mathbf{X}_n^0 ; the input and output in this paper are represented by a $N_x \times N_y \times 2$ -dimensional array that contains a two-dimensional velocity vector field. Once the CNN-AE is optimized, the encoding results of each flow field can be represented by the following equation:

$$\mathbf{Z}_n^0 = \text{encoder}(\mathbf{X}_n^0) \quad (3)$$

which is used as a low-dimensional feature quantity. This feature is considered to hold the information necessary for restoring a flow field, and it represents the field's spatial structure.

We propose that, in this paper, the CNN-AE is optimized using all of the available data that can be low-dimensionalized,

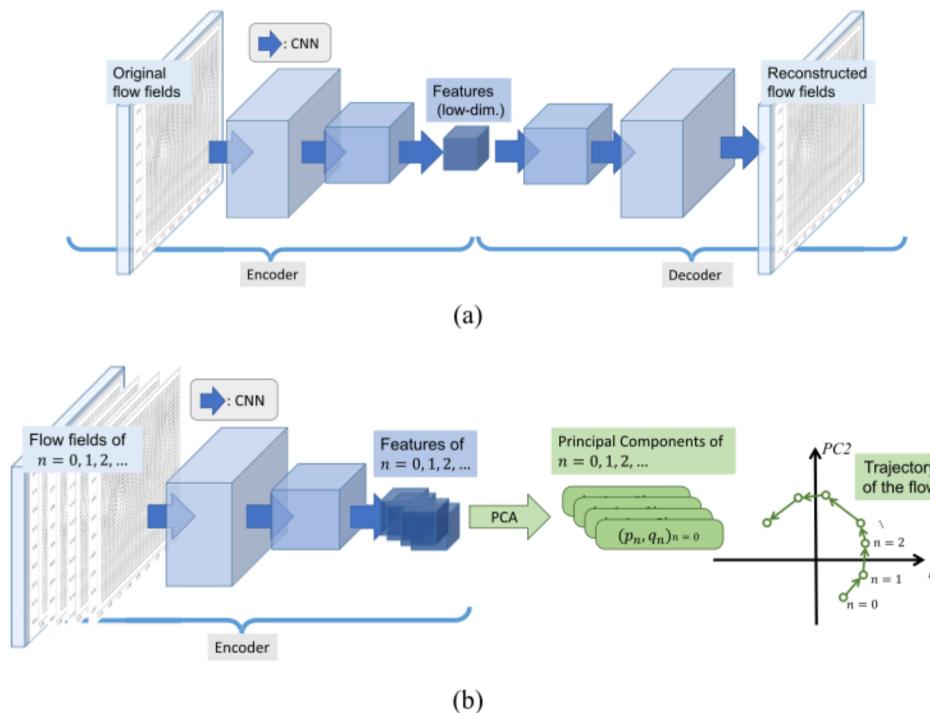


FIG. 4. Outline of our method. (a) Construction and optimization of CNN-AE. (b) Visualization based on the feature extracted by CNN-AE.

the flow field corresponds to not only θ to be visualized but all $\theta \in \Theta$. This is because CNN-AE can learn from a large amount of data, as its learning algorithm utilizes mini-batch learning; this is an advantage of deep learning methods not available to POD. In addition, since the time required for a CNN-AE to be optimized is long, it is time-consuming to optimize CNN every time the analysis target changes. Nevertheless, learning from the entirety of the data simultaneously rather than from just a sub-section of it is considered to be efficient.

By applying the above CNN-AE method, dimension reduction is performed and the spatial structure of the flow field at each time period is represented by low-dimensional features. We transform each instantaneous flow field, \mathbf{X}_n^θ , into a feature, \mathbf{Z}_n^θ using the optimized CNN-AE following Equation 3.

C. Visualization of spatio-temporal structures

The visualization method of the spatio-temporal structures of unsteady flows is an extension of the method described in Section II B. This visualization is performed on the spatial features obtained by the CNN-AE method proposed above, and it can be said to represent the spatio-temporal structure of a flow.

The features \mathbf{Z}_n^θ obtained as described in the previous subsection are projected into two-dimensional values to be visualized. We propose to use PCA here both because the original method for dynamic network²⁷ used it and because it has been used as POD in the field of fluid dynamics.

We propose that the PCA is performed only for the features of flows whose trajectories are actually represented, with features corresponding to the interested θ (i.e. user-selected sets of (Re, α)). This is because it is considered that taking into account data that will not be compared will worsen the quality of the visualization because those data are unrelated to what is intended to visualize. Furthermore, all of the data would need to be stored in memory for a linear algebra calculation of PCA. It is unsuitable for analysis of data cannot be on computer memory. For these reasons, it is considered difficult and unnecessary for PCA to be used on the entirety of the data; as such, it is only used on a limited part. Note that all $\theta \in \Theta$ is used in the optimization of the CNN-AE.

PCA is done using linear algebraic operation. Let us define a $N_t \times M$ -dimensional matrix that collect all the features to be visualized

$$\mathbf{H}^\theta = [\mathbf{Z}_1^\theta, \mathbf{Z}_2^\theta, \dots, \mathbf{Z}_{N_t}^\theta]^T. \quad (4)$$

Then we apply singular value decomposition to the matrix and obtain matrices $\mathbf{U}^\theta, \mathbf{S}^\theta, \mathbf{V}^{\theta T}$ as

$$\mathbf{H}^\theta = \mathbf{U}^\theta \mathbf{S}^\theta \mathbf{V}^{\theta T}. \quad (5)$$

Here, the diagonal matrix \mathbf{S}^θ is arranged so that its elements are aligned in descending order of absolute value. In this section, the proposed method is described by the first principal components p_n^θ and the second principal components q_n^θ . These are defined as

$$p_n^\theta = U_{n,1}^\theta S_{1,1}^\theta, \quad q_n^\theta = U_{n,2}^\theta S_{2,2}^\theta. \quad (6)$$

Finally, visualization is performed. By continuously plotting the first and second principal components p_n^θ, q_n^θ as values for horizontal and vertical axis respectively, the time series of the unsteady flow fields are represented as a trajectory. Schema of these procedures are shown in Figure 4(b). Each of the displayed points represents the spatial structure of the flow at certain times, and their trajectories represent the spatio-temporal structure of the flow.

IV. EXPERIMENTS AND RESULTS

To experimentally evaluate the proposed method, we applied it to unsteady flows around a two-dimensional airfoil.

The calculation method we used was the same as that used in Ref. 32. In this paper, NACA0012 was used as the two-dimensional airfoil, and O-type grid was used. An example of computational grid used is shown in gray in Figure 3. The chord length was set to 1, and the leading edge of the airfoil was set at $(-0.5, 0)$. The outer region is a circle with a radius of 50 centered on the origin of the coordinates. The number of grid points used was 200 in the radial direction and 360 in the circumferential direction. The airfoil was accelerated from a static state at $t = 0$ to $t = 5$ before being moved at a constant speed of $U_\infty = 1$. The calculation time step used was $dt = 0.005$, and an unsteady calculation was performed until $t = 100$ for 20,000 steps.

The parameters used for the data sampling were $x_s = 0.75$, $y_s = -1.0$, $x_e = 2.75$, $y_e = 1.0$, $N_x = 40$, and $N_y = 40$. The data sampling area is shown in red in Figure 3. Data sampling was conducted at five-step intervals and $\Delta t = 0.025$. Therefore, we obtained $N_t = 4,000$ instantaneous flow fields for each calculation.

Under the above conditions, calculations were performed by changing the Reynolds number and the angle of attack. The Reynolds numbers used were 1,000, 2,000, 3,000, 5,000, 10,000, 20,000, and 30,000. The angles of attack used were $0-12^\circ$ with 1° increment. In this paper, we show the results for the case of the different angles of attack.

A. Construction and optimization of CNN-AE

We firstly constructed a CNN-AE for low-dimensionization of flow fields. The structure of the CNN-AE used in the experiment is shown in Table I. The input and output layers were $40 \times 40 \times 2 = 3$, 200-dimensional arrays because the multi-dimensional data of each flow field were introduced to them. Furthermore, in this paper, the Pool-3 layer, $M = 5 \times 5 \times 8 = 200$ -dimensional quantity, was used as a low-dimensional feature, \mathbf{Z} , by the CNN-AE.

In Table I, the $x \times y \times d$ convolution indicates that a linear convolution with a kernel size of $x \times y$ is performed d times. Each convolution kernel is a parameter of the CNN-AE that will be optimized. ReLU represents an activation operation that applies to the ReLU function, as highlighted by the following equation:

$$\text{ReLU}(x) = \max(0, x) \quad (7)$$

TABLE I. Structure of the CNN-AE.

| Name | Shape | Operation performed |
|---------------|--------------------------|------------------------------------|
| Input | $40 \times 40 \times 2$ | — |
| Conv-1 | $40 \times 40 \times 16$ | $3 \times 3 \times 16$ convolution |
| Activate-1 | $40 \times 40 \times 16$ | ReLU activation |
| Pool-1 | $20 \times 20 \times 16$ | 2×2 maxpool |
| Conv-2 | $20 \times 20 \times 8$ | $3 \times 3 \times 8$ convolution |
| Activate-2 | $20 \times 20 \times 8$ | ReLU activation |
| Pool-2 | $10 \times 10 \times 8$ | 2×2 maxpool |
| Conv-3 | $10 \times 10 \times 8$ | $3 \times 3 \times 8$ convolution |
| Activate-3 | $10 \times 10 \times 8$ | ReLU activation |
| Pool-3 | $5 \times 5 \times 8$ | 2×2 maxpool |
| Conv-4 | $5 \times 5 \times 8$ | $3 \times 3 \times 8$ convolution |
| Activate-4 | $5 \times 5 \times 8$ | ReLU activation |
| Upsample-4 | $10 \times 10 \times 8$ | 2×2 upsampling |
| Conv-5 | $10 \times 10 \times 8$ | $3 \times 3 \times 8$ convolution |
| Activate-5 | $10 \times 10 \times 8$ | ReLU activation |
| Upsample-5 | $20 \times 20 \times 8$ | 2×2 upsampling |
| Conv-6 | $20 \times 20 \times 16$ | $3 \times 3 \times 16$ convolution |
| Activate-6 | $20 \times 20 \times 16$ | ReLU activation |
| Upsample-6 | $40 \times 40 \times 16$ | 2×2 upsampling |
| Output | $40 \times 40 \times 2$ | $3 \times 3 \times 2$ convolution |

This operation is applied to each value of a layer individually. Due to the non-linearity of the ReLU function, the CNN-AE gains a non-linear transformation. A pooling operation, shown as $x \times y$ maxpool in Table I, divides a layer in a small area of $x \times y$ and holds only the maximum value in that small area. This operation spreads the relevant area to convolve and has the effect of moving from a micro-viewpoint to a macro-viewpoint. Conversely, $x \times y$ upsampling operation copies the each quantity of the layer to a small area of $x \times y$ and plays a role in both returning the viewpoint to a microscopic one and increasing the layer dimension.

Additionally, for the loss function, which measures the difference between the values input into and output by the CNN-AE, the sum of the absolute errors

$$D(\mathbf{X}, \tilde{\mathbf{X}}) = \sum |\mathbf{X} - \tilde{\mathbf{X}}| \quad (8)$$

is used. In this equation, the sum is taken elementwise. Note that the average reconstruction error E is calculated by

$$E = \frac{D(\mathbf{X}, \tilde{\mathbf{X}})}{N_x \times N_y \times 2}. \quad (9)$$

Optimization of the CNN-AE was performed by using stochastic gradient descent method, which is commonly used for optimization of neural networks. An Adam algorithm³³ was used to determine the learning rate, and Keras,³⁴ which is a Python library developed for deep learning, was used for the implementation and optimization of the CNN-AE.

An example of flow fields being reconstructed by the constructed CNN-AE is shown in Figure 5. We showed the flow under $Re = 10,000$ and $\alpha = 10^\circ$. In this figure, the top row shows the original flow fields, the center shows the reconstructed ones, and the bottom shows the point-wise error of the reconstruction. In this figure, B, D, E, and F corresponds to the time period shown in Table II. In the plot of the instantaneous field, each velocity vector is indicated by an arrow, and vorticity distribution is visualized by color. Note that the vorticity is computed by using the sampled or reconstructed velocity field. Hereafter, instantaneous flow fields are shown in the same manner. Visually there is little difference between the original flow fields and the reconstructed flow fields. In fact, as shown at the bottom, the error of reconstruction was less than 0.05 at most positions. In addition, the time-series graph of the average reconstruction error for $\alpha = 10^\circ$ and 12°

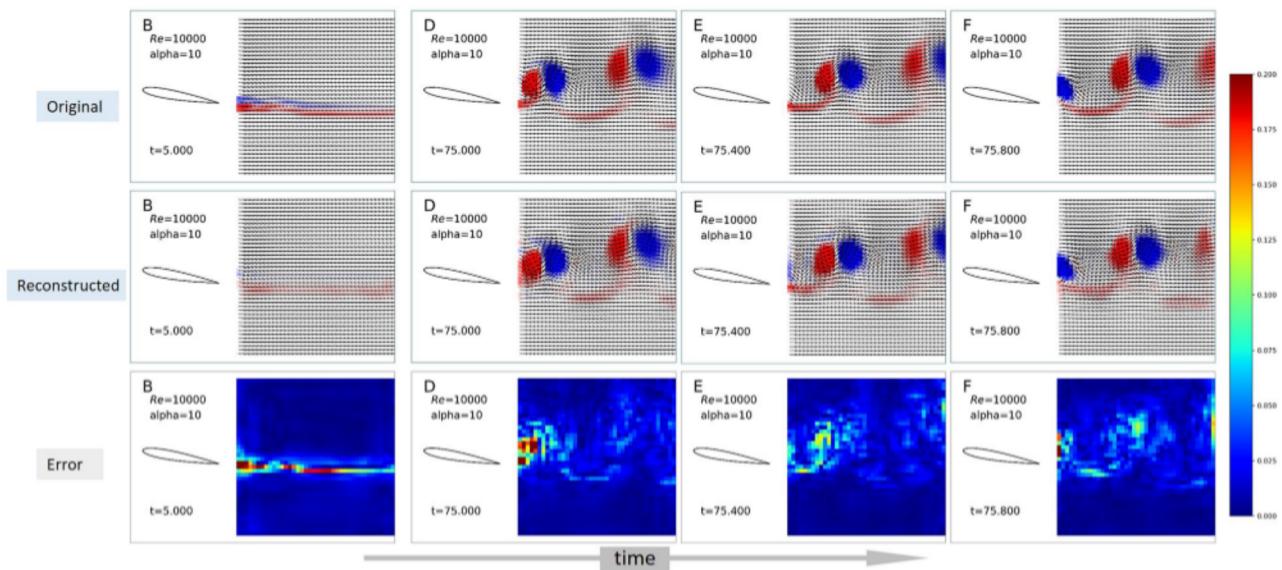
**FIG. 5.** Example of flow fields reconstruction by the CNN-AE. $Re = 10,000$, $\alpha = 10^\circ$.

TABLE II. Time period and the condition of flow.

| Name | Time | Condition |
|------|--------|-----------------------------------|
| A | 1.500 | Initial stage of the acceleration |
| B | 5.000 | End of the acceleration |
| C | 9.000 | First vortex structure |
| D | 75.000 | Quasi-periodic structure 1 |
| E | 75.400 | Quasi-periodic structure 2 |
| F | 75.800 | Quasi-periodic structure 3 |

under $Re = 10,000$ is shown in [Figure 6](#). As shown in [Figure 6](#), when the initial vortex reaches in this data sampling area, the reconstruction error become larger compared to the other time points. Since the flow field formed by the initial vortex appears only in this time section, it is considered that the error increases when learning by CNN-AE is performed with the entire data. In fact, the average error in E for all available flow fields was 9.6×10^{-3} , which was 0.96 % of the main

flow velocity, U_∞ . Therefore, it can be said that the low-dimensional representation of the CNN-AE sufficiently retains the spatial information of the flow fields.

B. Visualization of the spatio-temporal structure of an unsteady flow

Nextly, by using our method, we visualized the spatio-temporal structure of an unsteady flow that we had calculated with specific parameters. 200-dimensional features were obtained for each flow by the CNN-AE and PCA was performed on the features. The first and second principal components were plotted on a display space.

The result of the visualization for a single unsteady flow is shown in the left of [Figure 7](#) (Multimedia view); the Reynolds number Re was 10,000 and the angle of attack α was 10° . In this figure, the corresponding parts are denoted from A to F for the time period shown in [Table II](#). The figures shown on the right side show the instantaneous velocity fields at each time.

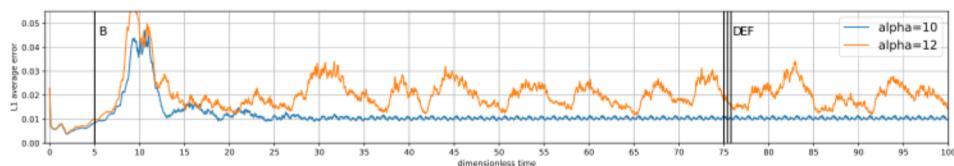


FIG. 6. Time-series graph of the average reconstruction error by the CNN-AE. $Re = 10,000$, $\alpha = 10^\circ$ and 12° .

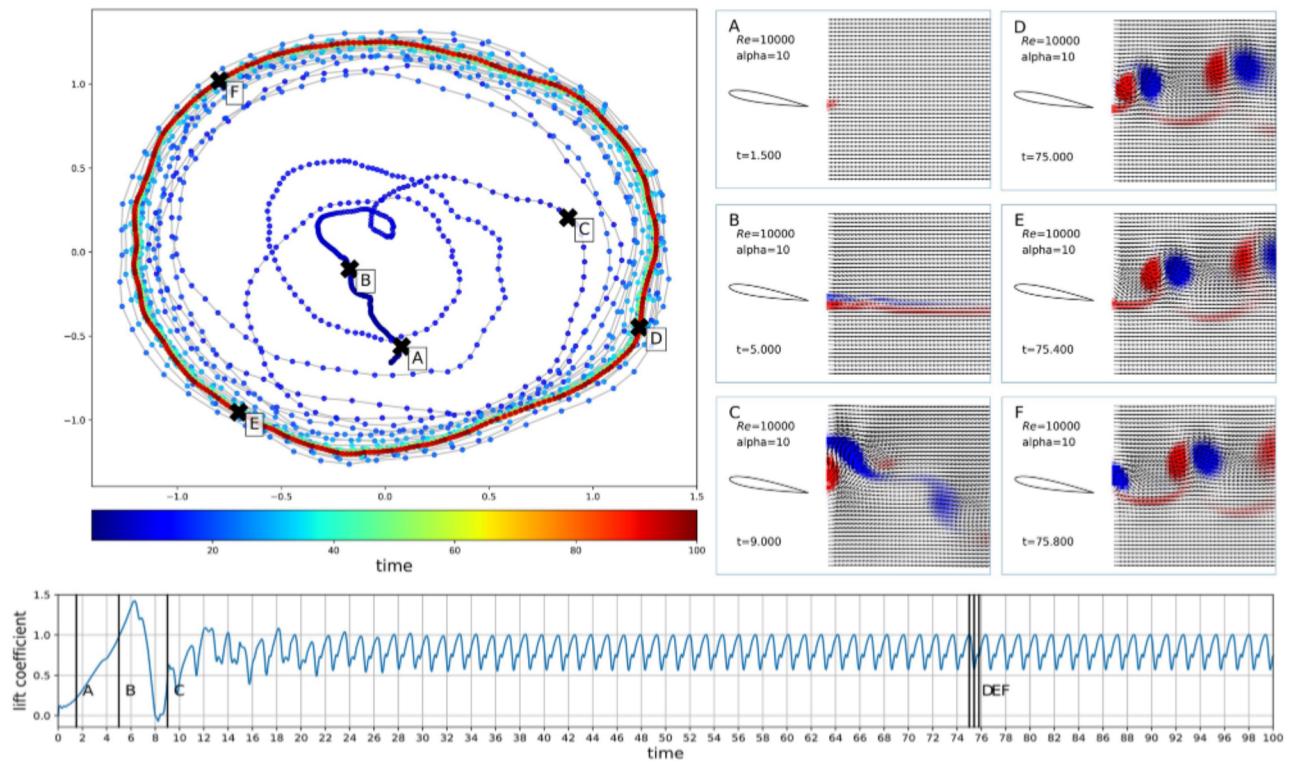


FIG. 7. Proposed visualization of spatio-temporal structure as a trajectory (left) and instantaneous flow fields (right). $Re = 10,000$, $\alpha = 10^\circ$. Multimedia view: <https://doi.org/10.1063/1.5067313.1>

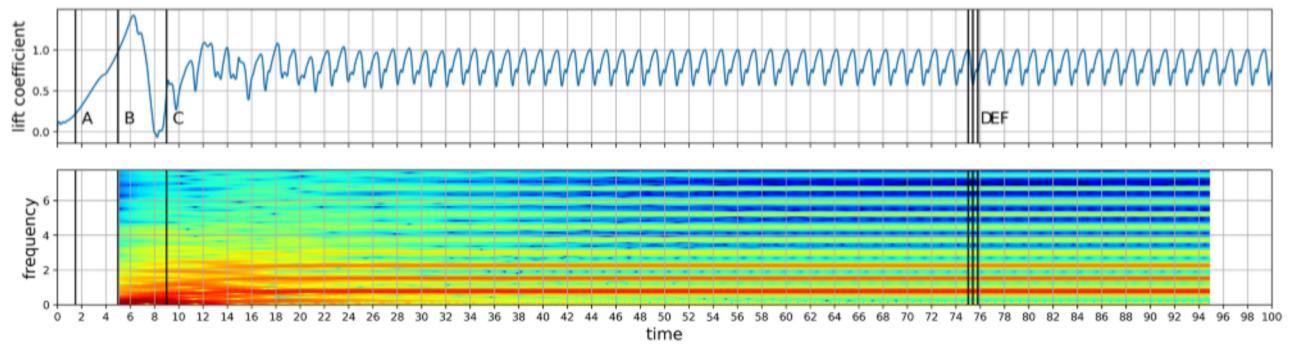


FIG. 8. Time-series graph (top) and spectrogram (bottom) of the lift coefficient. $Re = 10,000$, $\alpha = 10^\circ$.

In the figure containing the visualization, the horizontal axis represents the first principal component of the PCA, and the vertical axis represents the second. Each dot in the figure represents the spatial structure of the flow field at a certain time. The colors of the points represent time, and although at the start the points are blue, as time progresses they become redder. The instantaneous velocity field at each time is shown by direct vector display method using arrows. Figure 8, meanwhile, shows a time-series graph of the lift coefficient of the airfoil, which is often used as a rough visualization of temporal behavior.

Points A and B in Figure 7 (Multimedia view) represent the times at which the airfoil gradually accelerated from a stationary state at the start of the experiment ($t = 0$). In this parameter settings of the flow, an initial vortex continues to evolve even after the end of acceleration at $t = 5$, and the lift increases. At point C, the initial vortex appears in the data sampling area. Following this, the vortices are quasi-periodically generated. The points D–F were obtained by selecting three points for roughly one cycle of the cyclic structure in the vicinity of $t = 75$.

In the trajectory shown in Figure 7 (Multimedia view), which represents the low-dimensional feature induced by the

proposed method, there is a straight structure at the accelerating part. After that, it can be seen from this figure that the trajectory converges into a quasi-periodic cyclic structure. From this figure, we can estimate that the flow followed such a structural change.

It has been difficult to ascertain the temporal behavior of an unsteady flow using classic visualization methods; this was typified by the visualization shown at the right side in Figure 7 (Multimedia view), as it was difficult to confirm the presence or absence of periodicity in the flow from these figures. In order to estimate temporal behavior, therefore, we often referred to the time series graph of a specific physical quantity, such as that shown in Figure 8. The figure below is a visualization result of spectrogram obtained at each time using the window function. In this paper, the time interval at a certain time t is $[t - \tau, t + \tau]$, the hamming window is used for the window function, and we set τ to 5.12. As shown in Figure 8, this method did not show the spatial structure.

Here we compare our proposed method with two conventional methods (POD and t-SNE³¹). Trajectory in some characteristic space obtained by the conventional methods is demonstrated in Figure 9. The results from POD and t-SNE are shown in Figure 9(a) and 9(b) respectively. Compared to

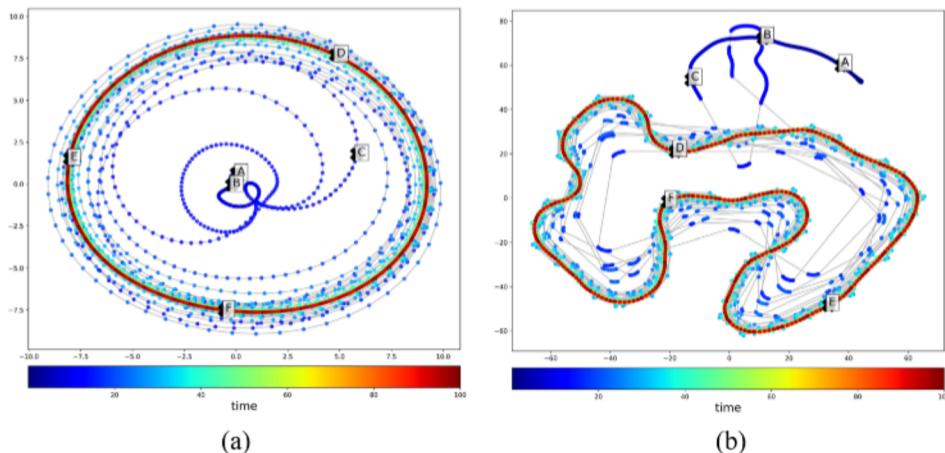


FIG. 9. Trajectory obtained by two conventional methods. $Re = 10,000$, $\alpha = 10^\circ$. (a) POD (b) t-SNE.

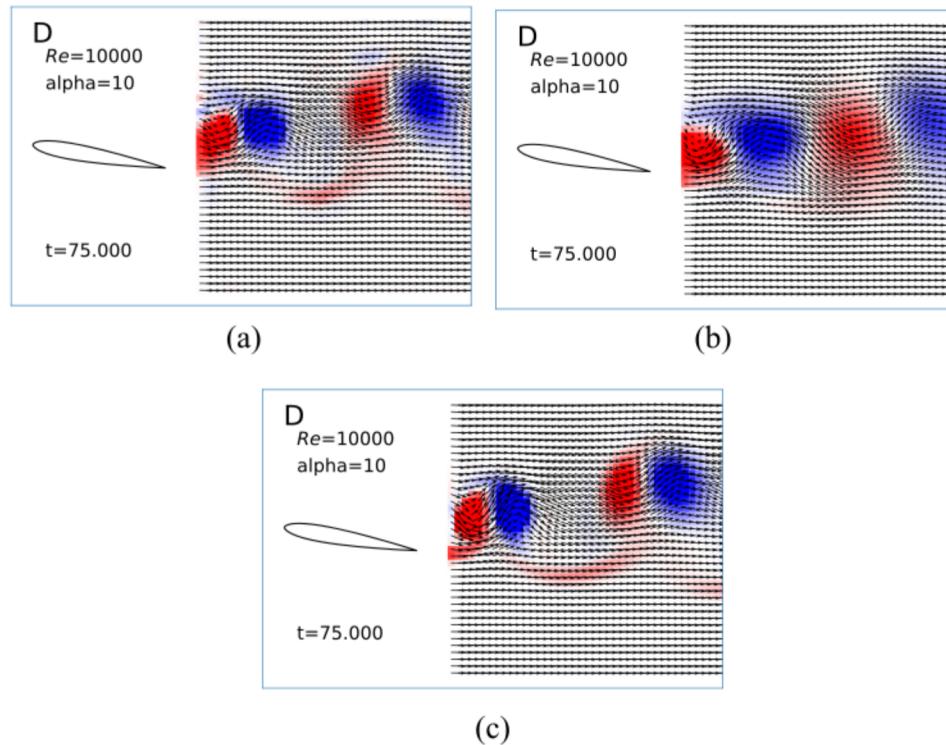


FIG. 10. Example of reconstruction from the information of the trajectory. $Re = 10,000$, $\alpha = 10^\circ$, $t = 75.0$. (a) Proposed method (AE+PCA) (b) POD (2 modes) (c) Original.

Figure 7 (Multimedia view), it is found that the trajectory in Figure 9(a) is a smooth elliptical orbit. This may be considered to be due to the linearity of POD. In the case of t-SNE, some discontinuity appear in the trajectory and the orbit becomes an irregular shape as shown in Figure 9(b). Also, the trajectory changes greatly due to a slight difference in the value of the parameter. Therefore, it is quite difficult to interpret the result from t-SNE. This is known as a weak point of t-SNE.

In contrast to these methods, in the visualization made using the proposed method, the spatio-temporal structure of the flow field can be seen to asymptotically shift from a linear structure to a quasi-periodic structure. Although the shift mode reported by Noack et al.³⁵ is effective in capturing transient physics, at the present time, the mode cannot be indicated by our method. In future work, we try to show this mode.

In addition, an example of the reconstructed flow fields from our method and the POD dominant modes are shown in Figure 10(a) and 10(b), respectively. In our method, the flow field was reconstructed using both the principal vector in PCA and the decoder function. The corresponding original flow field is demonstrated in Figure 10(c). It is shown that the result from our method is closer to the original flow field.

V. APPLICATION TO COMPARISON

Finally, we applied our method to comparison of the spatio-temporal structures of different flow fields. We firstly

show the result of comparison between two unsteady flows, and then we proceed to comparison among many unsteady flows.

A. Comparison of two unsteady flows

Firstly, flows for the angles of attack α of 10° and 12° under Reynolds number Re of 10,000 are compared. In order to compare fully developed flows, we handled the flow only after $t = 50$, $n = 2,001-4,000$.

We used the same CNN-AE optimized in the experiment described in the previous section. Since we have optimized CNN-AE for all available flow fields, it can be reused and there was no need to re-optimization. PCA is performed as almost the same way as when dealing with a single flow. The difference is construction of the collection matrix of the features H . This time, we put all the features to be compared into the matrix H as

$$H = [\dots, Z_n^{\theta_1}, \dots, Z_n^{\theta_2}, \dots]^T. \quad (10)$$

The principal components correspond to each features (or flow field) can be obtained by simply performing the same operation to this matrix.

Figures 11 and 12 show the result of the visualization for the flow of $Re = 10,000$ and $\alpha = 12^\circ$ and the time-series graph of the lift coefficient respectively. In this case, we use the first and second principal components for comparison with the results shown above. Comparing Figure 11 with Figure 7

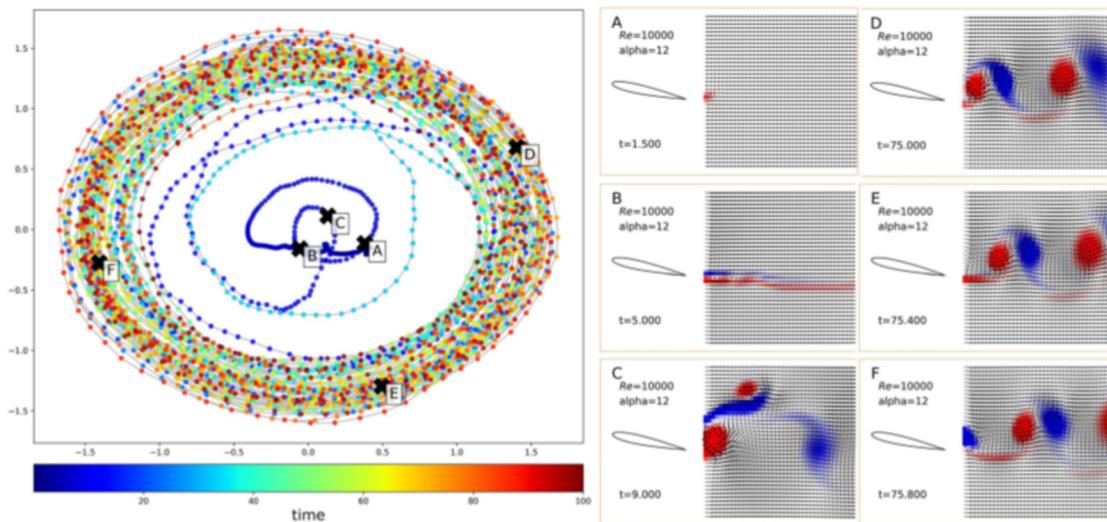


FIG. 11. Proposed visualization of spatio-temporal structure as a trajectory (left) and instantaneous flow fields (right). $Re = 10,000$, $\alpha = 12^\circ$.

(Multimedia view), we can find that the trajectory does not converge to a periodic structure in Figure 11. Although it may be considered that this result corresponds to the difference of time-variation of the lift coefficient (Figures 8 and 12), it is difficult to briefly show the difference of flow fields with different angle of attack. On the other hand, from our method, the difference of the flow field itself can be expressed in the same feature space. However, the details of the difference are not clear. To clarify this difference, we use the third order principal components that

$$r_n^\theta = U_{n,3}^\theta S_{3,3}^\theta. \quad (11)$$

The result of the comparison is shown in Figure 13 (Multimedia view). The representation in this figure conforms to that of Figure 7 (Multimedia view), but the color represents the angle of attack; temporal information is not represented in this figure. The visualization of temporal behavior by our

method is shown at the upper left side. The trajectory of the flow at $\alpha = 10^\circ$ is shown in blue and that at $\alpha = 12^\circ$ is shown in orange.

In the upper right of the figure, for $\alpha = 10^\circ$, the instantaneous fields at the times D, E and F in Table II are shown. For $\alpha = 12^\circ$, the points D, E and F at the 12 degrees trajectory that close to each corresponding point in the 10 degree trajectory are shown.

The trajectory of $\alpha = 10^\circ$ forms a certain closed curve in this three-dimensional feature space. On the other hand, the trajectory of $\alpha = 12^\circ$ behaves complicatedly around a cylindrical curved surface. In this case, in order to show the difference in the structure of flow between $\alpha = 10^\circ$ and $\alpha = 12^\circ$, the third principal component plays an important role. Although it is a future work to clarify the meaning of the third component, we can clearly show the difference of the two unsteady flow fields.

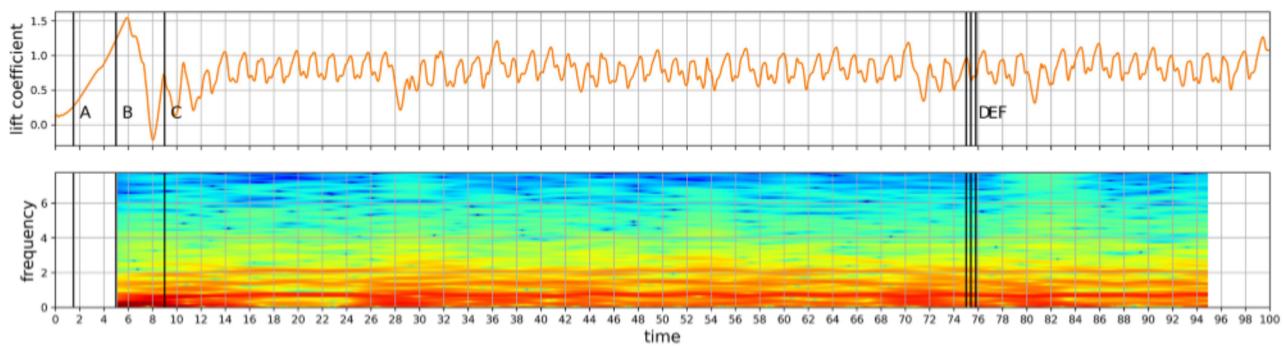


FIG. 12. Time-series graph (top) and spectrogram (bottom) of the lift coefficient. $Re = 10,000$, $\alpha = 12^\circ$.

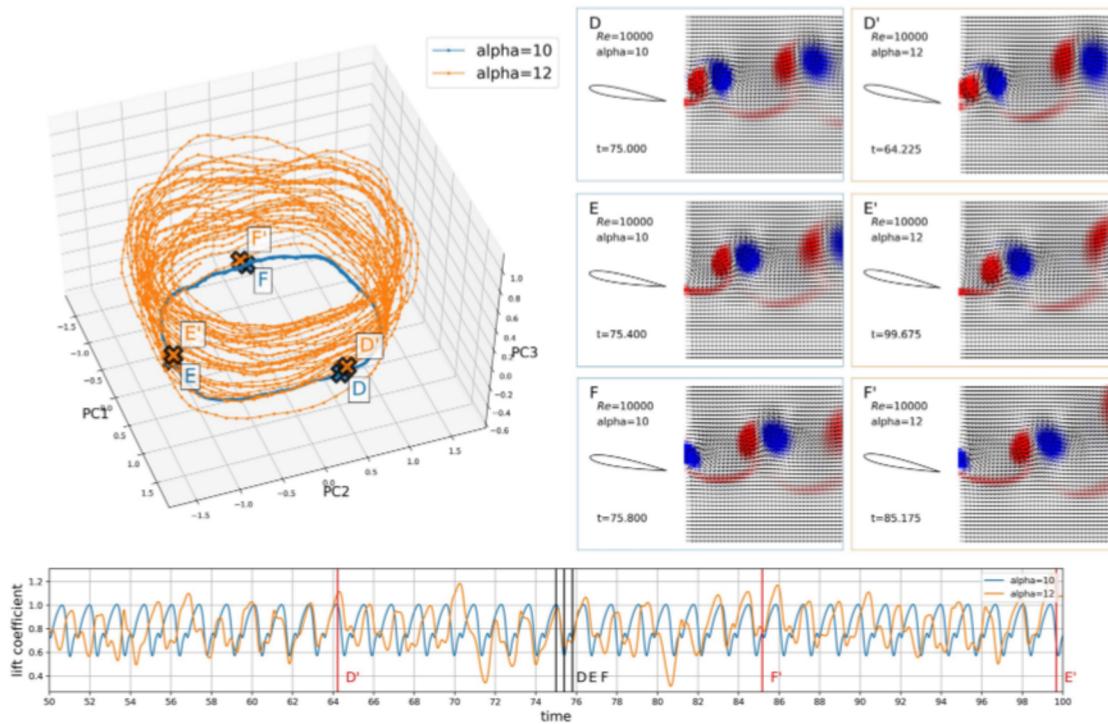


FIG. 13. Comparison between two unsteady flows at $Re = 10,000$. Blue indicates angle of attack $\alpha = 10^\circ$ and orange indicates $\alpha = 12^\circ$. Multimedia view: <https://doi.org/10.1063/1.5067313.2>

B. Comparison among many unsteady flows

Nextly, flows for the angles of attack α of $0\text{--}12^\circ$ under Reynolds number Re of 10,000 are compared. To compare the trajectories of fully developed flows, the flows after $t = 50$ were used.

The trajectories of the flow fields for every angle of attack are shown in Figure 14. The representation in the figure conforms to that of top left side of Figure 13 (Multimedia view).

In Figure 14, it can be seen that the trajectory of each flow has a different position and shape depending on the angle of

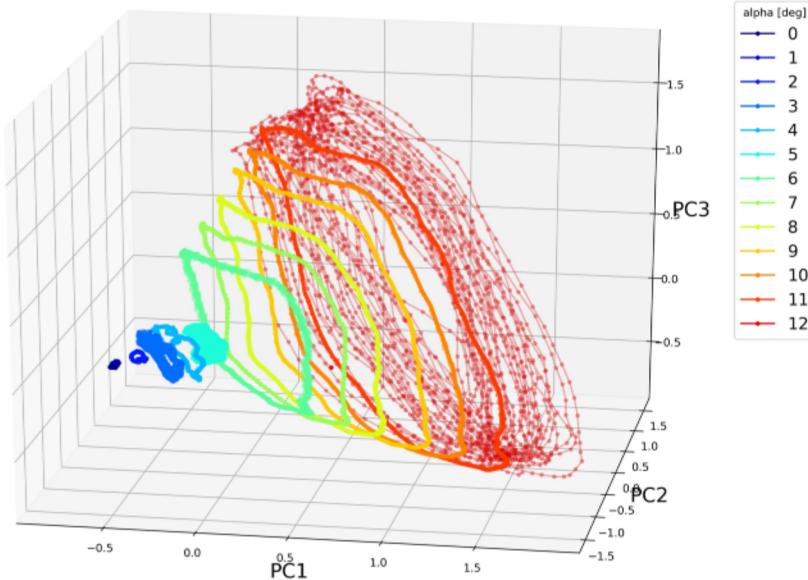


FIG. 14. Comparison among many unsteady flows at $Re = 10,000$. We varied the angle of attack in $\alpha = 0^\circ\text{--}12^\circ$.

attack. A flow with a small angle of attack is indicated on the left-hand side. With regard to the size of the trajectory, a flow with a small angle of attack can be seen to only cover a small area, while a flow with a large angle of attack covers a large area.

Furthermore, when comparing the shapes of the trajectories, those of $\alpha = 0\text{--}2^\circ$ are almost stationary; whereas those of $\alpha = 3\text{--}11^\circ$ have cyclic structures; $\alpha = 12^\circ$, meanwhile, does not have a quasi-periodic structure. The shapes of the trajectories of $\alpha = 3\text{--}11^\circ$ are different. More specifically, the shapes of the trajectories of $\alpha = 3\text{--}4^\circ$ are similar, whereas the shape of $\alpha = 5^\circ$ appears to be far more compact; the reason for why this is the case is unknown but this may indicate some physical phenomena. Although $\alpha = 6\text{--}11^\circ$ have similar shapes, they are not exactly the same as each, and it can be seen that their shapes gradually change as α increases.

VI. CONCLUSION

In this paper, we described the difficulty in handling time series data about unsteady flow fields, and we proposed a method for visualizing and comparing the spatio-temporal structure of these fields.

The proposed method consists of two stages: a mapping of the spatial structure of instantaneous flow fields to a low-dimensional feature space and a time series representation of the mapped data in the display space. For the mapping to the low-dimensional feature space, we proposed a method using CNN-AE, which is a deep learning technique. Furthermore, to show how these fields change over time, we proposed that a visualization method used for dynamic networks be extended to unsteady flow fields.

We applied the proposed method to unsteady flows around a two-dimensional airfoil, and we demonstrated that it could briefly represent the changes in the spatial structure of the unsteady flow field over time. This method was demonstrated to also be able to visualize changes in the quasi-periodic state of the flow when the angle of attack of the airfoil was changed. We demonstrated that it is possible to provide a visual representation of unsteady flow fields, which other quantitative methods have difficulty doing and therefore comparing.

In future, we intend to build upon the work in this paper by optimizing the parameters used for the CNN-AE and clarifying the physical meaning of the coordinate axes in the low-dimensional feature space.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support provided by JSPS KAKENHI Grant Number 16K14500.

REFERENCES

- ¹N. Ohno and A. Kageyama, "Region-of-interest visualization by CAVE VR system with automatic control of level-of-detail," *Computer Physics Communications* **181**, 720–725 (2010).
- ²B. Haddadi, C. Jordan, and M. Harasek, "Cost efficient CFD simulations: Proper selection of domain partitioning strategies," *Computer Physics Communications* **219**, 121–134 (2017).
- ³D. Tutunăea, M. Bica, and A. Dima, "Computational study of the unsteady flow structures around two vehicles," *Journal of Industrial Design & Engineering Graphics* **9** (2014).
- ⁴R. Volpe, V. Ferrand, A. Da Silva, and L. Le Moine, "Forces and flow structures evolution on a car body in a sudden crosswind," *Journal of Wind Engineering and Industrial Aerodynamics* **128**, 114–125 (2014).
- ⁵N. Agre, S. Childress, J. Zhang, and L. Ristroph, "Comparative flow visualization for steady and unsteady motions of a disk through a fluid," *Physics of Fluids* **27**, 091103 (2015).
- ⁶Y. A. Yusoff, F. Mohamad, M. S. Sunar, and A. Selamat, "Flow visualization techniques: A review," *Trends in Applied Knowledge-Based Systems and Data Science* (Springer International Publishing, 2016), pp. 527–538.
- ⁷T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen, "Over two decades of integration-based, geometric flow visualization," in *Computer Graphics Forum*, Vol. 29 (Wiley Online Library, 2010) pp. 1807–1829.
- ⁸C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).
- ⁹K. Taira, S. L. Brunton, S. T. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, "Modal analysis of fluid flows: An overview," *AIAA Journal* **55**, 4013–4041 (2017).
- ¹⁰C. W. Rowley and S. T. Dawson, "Model reduction for flow analysis and control," *Annual Review of Fluid Mechanics* **49**, 387–417 (2017).
- ¹¹A. Hadjighasem, D. Karrasch, H. Teramoto, and G. Haller, "Spectral-clustering approach to Lagrangian vortex detection," *Physical Review E* **93**, 063107 (2016).
- ¹²S. Jeong, B. Solenthaler, M. Pollefeys, M. Gross et al., "Data-driven fluid simulations using regression forests," *ACM Transactions on Graphics (TOG)* **34**, 199 (2015).
- ¹³Z. Bai, S. L. Brunton, B. W. Brunton, J. N. Kutz, E. Kaiser, A. Spohn, and B. R. Noack, "Data-driven methods in fluid dynamics: Sparse classification from experimental data," *Whither Turbulence and Big Data in the 21st Century?* (Springer, 2017), pp. 323–342.
- ¹⁴A. P. Singh, S. Medida, and K. Duraisamy, "Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils," *AIAA Journal* **55**, 2215–2227 (2017).
- ¹⁵G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science* **313**, 504–507 (2006).
- ¹⁶A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105.
- ¹⁷J. N. Kutz, "Deep learning in fluid dynamics," *Journal of Fluid Mechanics* **814**, 1–4 (2017).
- ¹⁸Y. Lee, H. Yang, and Z. Yin, "PIV-DCNN: Cascaded deep convolutional neural networks for particle image velocimetry," *Experiments in Fluids* **58**, 171 (2017).
- ¹⁹J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics* **807**, 155–166 (2016).
- ²⁰D. Stoecklein, K. G. Lore, M. Davies, S. Sarkar, and B. Ganapathysubramanian, "Deep learning for flow sculpting: Insights into efficient learning using scientific simulation data," *Scientific Reports* **7**, 46368 (2017).
- ²¹C. Galletti, E. Brunazzi, M. Yiannakis, and A. Paglianti, "Spectral and wavelet analysis of the flow pattern transition with impeller clearance variations in a stirred vessel," *Chemical Engineering Science* **58**, 3859–3875 (2003).
- ²²D. Faranda, F. M. E. Pons, B. Dubrulle, F. Daviaud, B. Saint-Michel, É. Herbert, and P. P. Cortet, "Modelling and analysis of turbulent datasets using auto regressive moving average processes," *Physics of Fluids* **26**, 105101 (2014).

- ²³J. L. Lumley, "The structure of inhomogeneous turbulent flows," *Atmospheric Turbulence and Radio Wave Propagation* **790**, 166–178 (1967).
- ²⁴C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics* **641**, 115–127 (2009).
- ²⁵H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology* **24**, 417 (1933).
- ²⁶I. Bright, G. Lin, and J. N. Kutz, "Classification of spatiotemporal data via asynchronous sparse sampling: Application to flow around a cylinder," *Multiscale Modeling & Simulation* **14**, 823–838 (2016).
- ²⁷S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk, "Reducing snapshots to points: A visual analytics approach to dynamic network exploration," *IEEE Transactions on Visualization and Computer Graphics* **22**, 1–10 (2016).
- ²⁸F. Beck, M. Burch, S. Diehl, and D. Weiskopf, "The state of the art in visualizing dynamic graphs," *EuroVis STAR* (2014), pp. 83–103.
- ²⁹M. Greilich, M. Burch, and S. Diehl, "Visualizing the evolution of compound digraphs with TimeArcTrees," in *Computer Graphics Forum*, Vol. 28 (Wiley Online Library, 2009) pp. 975–982.
- ³⁰Y. Frishman and A. Tal, "Dynamic drawing of clustered graphs," in *Proceedings of 2004 IEEE Symposium on Information Visualization* (IEEE, 2004) pp. 191–198.
- ³¹L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research* **9**, 2579–2605 (2008).
- ³²S. Shirayama and K. Kuwahara, "Flow past a sphere—Topological transitions of the vorticity field," *AIAA Journal* **30**, 349–358 (1992).
- ³³D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference for Learning Representations* (2015).
- ³⁴F. Chollet et al., "Keras," <https://github.com/fchollet/keras> (2015).
- ³⁵B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, and F. Thiele, "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake," *Journal of Fluid Mechanics* **497**, 335–363 (2003).