

Streamline Selection for Comparative Visualization of 3D Fluid Simulation Result

Shoko SAWADA^{*}, Takayuki ITOH^{*}, Takashi MISAKA⁺, Shigeru Obayashi⁺,
Tobias CZAUDERNA⁺⁺, and Kingsley STEPHENS⁺⁺

(^{*}) Ochanomizu University, (⁺) Tohoku University, (⁺⁺) Monash University
{shoko,itot}@itolab.is.ocha.ac.jp, misaka@edge.ifs.tohoku.ac.jp, obayashi@ifs.tohoku.ac.jp,
{Tobias.Czauderna, kingsley.stephens}@monash.edu

Abstract

Fluid dynamics simulation is often repeated while changing conditions, and therefore we need to compare a large amount of results. In order to compare results under different conditions, it is effective to overlap the streamlines generated from each condition in a single 3D space. Streamline is a curved line which represents a wind flow. This paper presents a technique to automatically select and visualize important streamlines suitable for comparison of the simulation results. In addition, we present an implementation to observe the flow fields in virtual reality spaces.

Keywords--- Visualization, Streamline, Virtual Reality

1. Introduction

These days, computational simulation has been applied in various fields owing to evolution of high-performance computers for large-scale calculations. Simulation and visualization play important roles especially in the research field of invisible fluid dynamics. Techniques on computational fluid dynamics (CFD) have been applied to numerically analyze and simulate phenomena of fluid with computers. Visualization using computer graphics (CG) is widely used as a methodology to analyze CFD simulation results.

Streamline is a popular technique to visualize vector fields obtained from three dimensional CFD simulation results. Streamline is a curved line whose tangents are equal to velocity vectors. By generating and displaying appropriate number of streamlines for each CFD simulation result, we can represent features of an entire flow field from a CFD simulation result. Also, we can intensively observe the interesting parts by displaying streamlines only in particular places in a CFD space. However, it is a complicated problem to automatically generate appropriate sets of streamlines, and therefore many techniques on this problem have been discussed.

Meanwhile, fluid dynamics simulation is often repeated while changing conditions. Therefore, we need to compare a large amount of simulation results. By comparing and analyzing these simulation results, we can understand fluid phenomena and improve the accuracy of fluid simulations. However, it is not easy for

users to compare these results by merely displaying images of them side by side. Comparative visualization methods for fluid have been therefore recently developed. While large number of comparative visualization techniques for scalar fields have been presented, we suppose it is still an open problem to develop effective comparative visualization techniques for vector fields.

We have developed a comparative visualization technique [1] for three-dimensional CFD simulation results by overlapping streamlines. This technique generates streamlines for each of volume datasets generated as CFD simulation results from the same seed positions. We can comparatively visualize the flow fields by overlapping the streamlines. However, this technique has not automated the streamline generation process; we needed to manually specify seed points of the streamlines.

This paper presents an automatic streamline selection method for comparative visualization of CFD simulation results. Specially, the technique firstly generates a considerable number of streamlines in the entire space targeted by the three-dimensional CFD simulation, then automatically selects and visualizes certain number of meaningful streamlines. We expect to effectively discover differences of the flow fields caused by changes of conditions.

Moreover, this paper presents a system for observing important streamlines elected by our method in a virtual reality (VR) space. It is often difficult to maintain visibility by just selecting streamlines and displaying by a usual 3D graphics tool, if important streamlines are concentrated in only a small portion of a CFD space. Also, fluid phenomena cannot be sufficiently recognized in principle, by using a 2D display while observing 3D fluid simulation results. We expect VR environments make it easier to compare and analyze 3D fluid simulation results more appropriately.

In the remainder of this paper, Section 2 lists related work. In Section 3, we discuss our proposed technique that selects important streamlines to compare in detail. Section 4 describe the results of our method. In Section 6, we present our conclusions and discuss future work.

2. Related Work

2.1. Comparative Visualization Method for Fluid

Study of comparative visualization has a long history [2]. Hattanda et al. presented a visualization method [3] which is used to compare EFD (experimental fluid dynamics) experiments and CFD simulations. Lampe et al. [4] presented another visualization technique which enables comparison among volume datasets which have different structures. The technique reconstructs the simulation spaces by bending orthogonal structures.

2.2. Streamline Selection

There have been many techniques on automatic streamline selection targeting a single CFD simulation result. Furuya et al. [5] presented a streamline selection technique which preserves high comprehensibility of flow fields by determining occlusion by isosurfaces in simultaneous visualization of isosurfaces and streamlines. Lee et al. [6] proposed a visualization method that recommends the best viewpoint where streamlines projected onto a screen are highly evaluated. They mentioned that bad choices of viewpoints may damage the comprehensibility of flow fields when too many streamlines are to be displayed. Ma et al. [7] proposed another method to select streamlines by calculating view-independent and view-dependent importance of streamlines applying with an interactive viewpoint manipulation mechanism.

2.3. Comparative Visualization using Streamline

Hattanda et al. [1] presented a technique that comparatively displays streamlines generated from two simulation results under different conditions in a single 3D space. This approach clarifies the differences of flow fields. Figure 1 is an example of visualizing two simulation results of the Haneda airport runway. Flow fields as the results of two CFD simulations under different conditions are visualized by streamlines drawn in pink and cyan. We can immediately find that the wind flows vary while changing the conditions.

However, this method requires to interactively set the seed points of streamlines one by one. As a result, operations for this method are complicated, and also, it is problematic that users need knowledge and experience for setting appropriate sets of seed points.

2.4. Visualization method using VR

Visualization for science and technology has a long history to apply VR technologies [8]. Indeed, VR has been positively applied to visualization of 3D vector fields [9]. For example, in order to improve the success rate of arterial transplantation, Forsberg et al. proposed a system [10] that simulates transplantation including blood flow in VR space. Coffey et al. proposed a visualization method [11] that realized compatibility

between overview and detail using 3D and multi-touch display technologies. The method makes it possible to observe the entire 3D data in the VR space and a specific cross section of the data on the display.

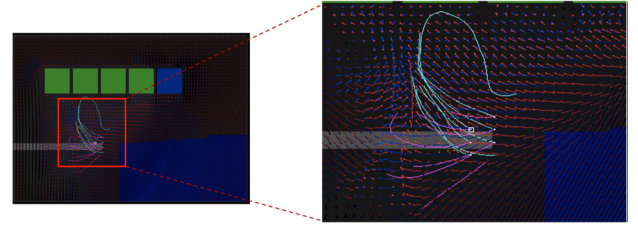


Figure 1: Result of streamlines generation by using method[1]. The green rectangles are the existing building. Simulations were executed with and without buildings at blue square positions. Streamlines generated in the simulation result mimicking before the building construction are drawn in pink and after the building construction are drawn in cyan.

3. Proposed Technique

This section introduces a processing flow of our automatic selection method to compare multiple 3D CFD simulation results. The section also proposes a system to observe the automatic selection results in a VR space.

In this section, we define “streamline pair” as the two streamlines generated at the same seed point for each of the two simulation results performed under different conditions. We suppose that all grid-points in the 3D space can be treated as seed points of the streamlines, and the streamlines are paired at all seed points. Therefore, the automatic streamline selection presented in this paper corresponds to automatic selection of appropriate sets of “streamline pairs”.

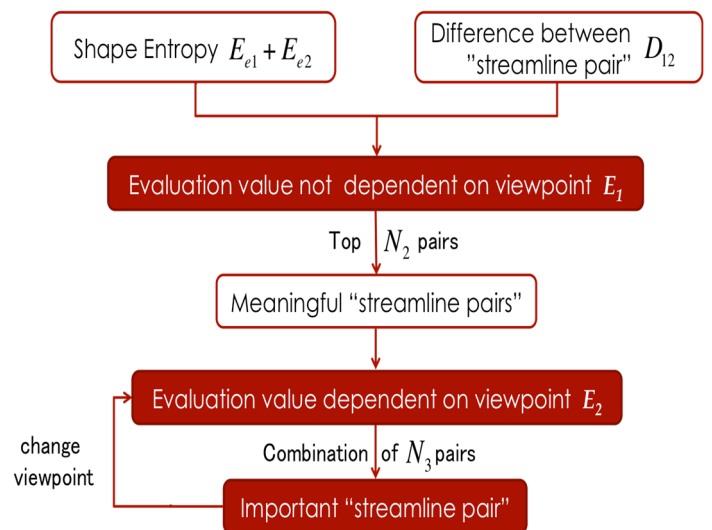


Figure 2: Processing flow for automatic selection

Figure 2 shows the processing flow of the automatic streamline selection. In this method, N_1 pairs of “streamline pairs” are firstly generated over the entire 3D space on trial. Our implementation generates N_1 pairs of “streamline pairs” while randomly selecting grid-points as seed points. Next, the following two values are calculated for each of the “streamline pairs”.

- shape entropy $E_{e1} + E_{e2}$
- difference between “streamline pair” D_{12}

From these two values, the method calculates a view-independent evaluation value E_1 . The method then stores the upper N_2 ($N_1 > N_2$) streamline pairs based on their E_1 values, where the stored streamline pairs are treated as meaningful to visualize. After that, the method calculates a view-dependent evaluation value E_2 for each of the stored streamline pairs, and selects the upper N_3 ($N_2 > N_3$) streamline pairs based on their E_2 values. Finally, the N_3 pairs are drawn in a 3D space as important streamline pairs. E_2 must be recalculated whenever a user changes the viewpoint. In the following description, we assume that streamlines are approximated as polygonal lines consisting of many connected short line segment.

3.1. View-independent evaluation for streamline pair selection

This section firstly describes the view-independent evaluation value E_1 . We just calculate E_1 as linear combination of shape entropy and difference between a streamline pair. The following is the formula to calculate E_1 :

$$E_1 = \alpha(E_{e1} + E_{e2}) + (1 - \alpha)D_{12} \quad (0 \leq \alpha \leq 1)$$

This value can be calculated in a preprocessing step executed only once when a CFD simulation result is given.

3.1.1. Shape Entropy

We applied the shape entropy defined by Ma et al. [8] which was originally defined to select streamlines from a single CFD simulation result. This definition works to preferentially select long and undulated streamlines, not short or straight streamlines. Here, short streamlines are less informative, and straight streamlines do not represent interesting characteristics of flow fields. Therefore, we did not select these types of streamlines initiatively.

Specifically, the following formula is used to calculate the information entropy:

$$E_{ei} = - \sum_x p(x) \log(x)$$

Since E_{e1}, E_{e2} can be calculated independently, we do not need to be conscious of streamlines being paired at this moment.

3.1.2. Difference between a pair of streamlines

Next, we describe the definition of the difference between a pair of streamlines D_{12} . We preferentially select a pair of streamlines if they are distant to each other.

We calculate D_{12} between a pair of streamlines from distances between vertices of the streamlines. First, we match vertices of a pair of streamlines one-by-one. Our

implementation simply identifies the vertex closest to the vertex in the other streamline to make pairs of vertices. We then calculate the distances between each of the matched two vertices. Finally, we calculate the average of these distance and treat it as D_{12} of a pair of streamlines.

Figure 3 illustrates the process of coupling vertices in a pair of streamlines.

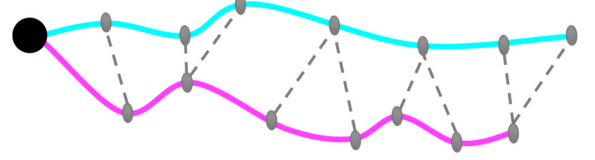


Figure 3: Coupling vertices to calculate the difference between a pair of streamlines. Pink curve line and cyan one are a pair of streamlines. Black dot is a seed point of this pair of streamlines, and gray dots are vertices of streamlines.

3.2. View-dependent evaluation for streamline pair selection

This section describes the view-dependent evaluation value E_2 . We aim to select a set of streamline pairs that minimizes their overlaps on a screen space. Like the shape entropy, there have been several studies on view-dependent streamline evaluations which were defined to select streamlines from a single CFD simulation result. In this study, we apply the view-dependent streamline evaluation value proposed by Furuya [6].

We display streamline pairs in the order of their importance. Streamlines which are significantly overlapped with already displayed streamlines will not be selected. Concretely, after the most important streamline pair is displayed, we will process the following steps in descending order of E_2 from the second place streamline pair.

1. Extract vertices of line segments constituting the current streamline.
2. Identify the nearest vertex of the streamline and calculate the distance to each vertex of the displayed streamlines.
3. Unselect the current streamline if it has more than a constant number of vertices of already displayed streamlines whose distances are less than the pre-defined threshold d .

This process must be repeated until N_3 streamline pairs are drawn.

3.3. Visualization in a VR space

This section describes a system to display the selected sets of streamlines in a VR environment. Even though our streamline selection method applies a view-dependent streamline evaluation, it may happen that the streamlines are concentrated in a certain part of a 3D simulation space, and comprehensibility of the visualization results cannot be therefore maintained. Also, observation of 3D simulation results on a 2D display

device may inevitably lead to a lack of understanding of flow dynamics due to deterioration of visibility. Therefore, we propose to display the streamline selection results in a VR environment in order to enable more accurate and detailed comparison of 3D data on a 3D display device.

We implemented the automatic streamline selection in Java. The selected streamlines can be displayed by this Java application as well (see Figure 7) but we have also implemented a display application in Unity for use in a VR environment, e.g., an Oculus Rift (see Figure 8).

Given fluid simulation results as regular grid datasets, our technique selects streamline pairs by the automatic selection method implemented in Java and stores the result as JSON files. Our implementation in Unity loads the JSON files and draws streamlines as connected cylinders as illustrated in Figure 4. It is easier to grasp the positional relationship of the streamlines in the depth direction while drawing cylinders. Figure 5(top) demonstrates the effectiveness of this implementation rather than drawing streamlines as planar lines as shown in Figure 5 (bottom).

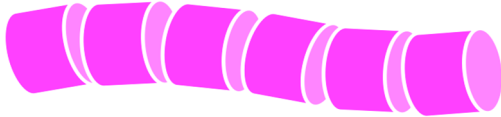
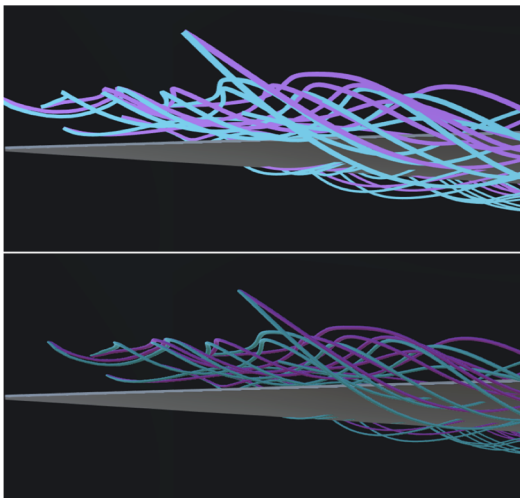


Figure 4: Example of a streamline drawn using connected cylinders



**Figure 5: Comparison of the styles to draw streamlines:
drawing as cylinders (top),
drawing as planar lines (bottom)**

Furthermore, we implemented the movement of the viewpoint (the camera in a VR environment) along a streamline selected by a user. We expect users to gain a better understanding by comparing streamlines experiencing the undulation of the flow. Users can also observe all its surroundings from the start point to the end point of the streamline.

The Unity application supports a variety of operations including fundamental viewing operations such as rotation, scaling, and panning, as well as

interactive streamline selection to move the viewpoint (the camera in a VR environment). We implemented all these operations for use with an Xbox controller, alternatively with a mouse and a keyboard can be used as well.

4. Example of Execution

4.1. Case Study

We applied the CFD simulation results of delta wing with different case angle of attack. A delta wing is an aircraft wing whose shape looks similar to the letter “Δ” of the Greek alphabet. As shown in Figure 6, the angle of attack is an angle between a chord line which is a straight line connecting the front end and the rear end of the wing and the flight direction. In other words, it is a value to show how long a body inclines for a flow.

In this paper, we compared two simulation results in the cases of 20 degrees and 27 degrees angle of attack. This section calls the former case “result A”, and the latter case “result B”. We also set parameters $N_1 = 10000$, $N_2 = 200$, and $N_3 = 20$.

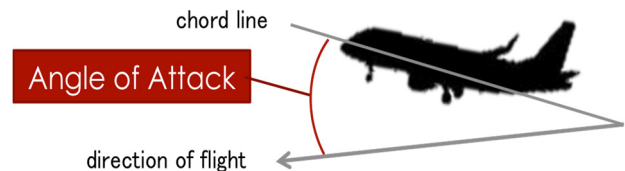


Figure 6: The red angle shows angle of attack.

4.2. Visualization Result

Figure 7 shows a visualization result displayed by our Java program. Streamlines in the simulation result in the case of 20 degrees are drawn in pink, and in the case of 27 degrees are drawn in cyan.

Figure 7 (top) shows a result with streamlines selected randomly, and Figure 7 (bottom) shows a result with streamlines selected by our method. A lot of short streamlines and straight streamlines were selected in Figure 7 (top). On the other hand, largely undulating streamlines were displayed using our method as shown in Figure 7 (bottom). This result demonstrates our method makes it easier to globally represent the entire flow field.

At the same time, we could find that this result depicted large undulation only by streamlines drawn in cyan. It is well-known that vortices occur behind a wing if the angle of attack becomes larger. The body of an airplane will be in a dangerous condition in this case. This is a significant knowledge to estimate the performance of airplanes.

Figure 8 shows a visualization result using the Unity application. A gray thin triangular prism visible in this figure is a modeled version of the delta wing. Because the position information of the wing is not described in

the dataset, this arrangement is not completely accurate. We felt stereoscopic effect with this visualization because this implementation draws streamlines not as lines but as 3D cylinders.

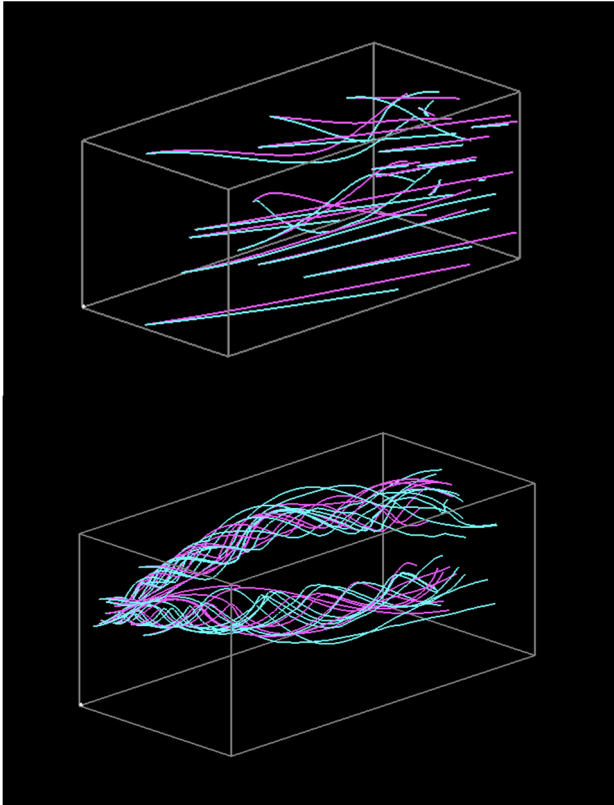


Figure 7: Visualization result using the Java application: streamlines selected randomly (top), streamlines selected by our method (bottom)

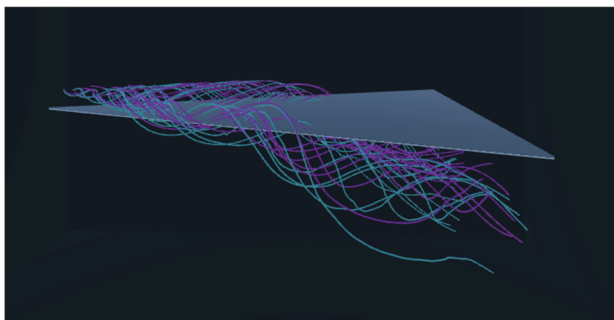


Figure 8: Visualization result using the Unity application

5. Conclusion and Future Work

This paper presented our method to select an appropriate number of streamline pairs in order to compare and visualize the results of two CFD simulations executed with changing conditions. This paper also presented a system to visualize the streamlines in a VR environment.

The presented method selects and keep a certain number of streamline pairs based on view-independent evaluation over the entire 3D space as a preprocessing.

When a user operates the viewpoint, the method selects an appropriate number of streamline pairs from the stored streamlines based on view-dependent evaluation. Then we save them to a JSON file and Unity reads it. Finally, we observed selected streamline pairs in a VR environment with the model of the target data.

With the improvement of computer performance and scientific computing technology, increasing number of simulations will be repeated more and more. Large amount of simulation results will be accumulated, and analysis schemes to compare the simulation results will be more important. We can clarify the differences caused by the condition change in CFD simulation results by using our method. We expect it will make easier to appropriately compare the simulation results.

Our potential future work include the following.

- 1) We simply calculate the linear combination of E_{e1} and E_{e2} as the definition of view-independent evaluation value E_1 . We have not conducted sufficient tests to verify whether this evaluation is truly balanced between representation of the entire flow field and emphasis of important parts where streamlines are significantly different. We want to proceed the tests on this point.
- 2) Our current implementation supports only orthogonal regular grids which divide the 3D space into the equally shaped and sized rectangular solids as CFD simulation results. In order to support more various fluid simulation results, we are extending the implementation so that unstructured grids can be applied.
- 3) We also would like to extend the implementation to support linked views with information visualization methods such as scatterplots. For example, we would like to implement a more interactive visualization mechanism to significantly control the threshold values for selecting important streamlines while browsing scatterplots.
- 4) We will implement interactive manipulation of streamline selection results in Unity.

References

- [1] K. Hattanda, A. Kuwana, T. Itoh, A Comparative Visualization for Flow Simulation of Airport Wind, NICOGRAPH International, 2015.
- [2] H. G. Pagendarm, F. H. Post, Comparative Visualization Approaches and Examples, in Visualization in Scientific Computing, Springer, 95-108, 1995.
- [3] K. Hattanda, T. Itoh, S. Watanabe, S. Kuchi-ishi, K. Yasue, A Flow Representation for EFD/CFD Integrated Visualization, NICOGRAPH International, 1234-1237, 2012.
- [4] O. D. Lampe, C. Correa, K. -L. Ma, H. Hauser, Curve-Centric Volume Reformation for Comparative Visualization, Transactions of Visualization and Computer Graphics, IEEE, Vol. 15, Issue 6, 1235-1242, 2009.
- [5] S. Furuya, T. Itoh, A Streamline Selection Technique for Integrated Scalar and Vector Visualization, The Journal of the Society for Art and Science, Vol. 8, 120-129, 2009.
- [6] T. -Y. Lee, O. Mishchenko, H. -W. Shen, R. Crawfis, View Point Evaluation and Streamline Filtering for Flow Visualization, Pacific Visualization, IEEE, 83-90, 2011.

- [7] J. Ma, C. Wang, C. Shene, Coherent View-Dependent Streamline Selection for Importance-Driven Flow Visualization, IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics., 865407-865407, 2013.
- [8] S. Bryson, Virtual reality in scientific visualization, Communications of the ACM, Vol. 39, Issue 5, 62-71, 1996.
- [9] A. Kageyama, N. Ohno, Interactive Three-Dimensional Visualization Software by Virtual Reality Technology, J. Plasma Fusion Res. Vol.84, No.11, 834-843, 2008.
- [10] A. Forsberg, D. Laidlaw, A. Van Dam, R. Kirby, G. Karniadakis, J. Elion, Immersive virtual reality for visualizing flow through an artery, In Proceedings of the Conference on Visualization, Computer Society Press, IEEE, 457-460, 2000.
- [11] D. Coffey, N. Malbraaten, T. Le, I. Borazjani, F. Sotiropoulos, D. F. Keefe, Slice WIM: A Multi-Surface, Multi-Touch Interface for Overview+Detail Exploration of Volume Datasets in Virtual Reality, I3D '11 Symposium on Interactive 3D Graphics and Games, 191-198, 2011.