Vibrations in Physical Systems 2020, 31, 2020319 https://doi.org/10.21008/j.0860-6897.2020.3.19

AI-based Method of Vortex Core Tracking as an Alternative for Lambda2

Maciej MAJCHRZAK
Konstrubowski Engineering Group
ul. 28 Czerwca 1956r. 406, 61-441 Poznań, Poland
maciej.majchrzak@konstrubowski.pl

Mateusz JAKUBOWSKI
Institute of Applied Mechanics, Poznan University of Technology
ul. Jana Pawła II 24, 60-965 Poznań, Poland
mateusz.ja.jakubowski@doctorate.put.poznan.pl

Roman STAROSTA
Institute of Applied Mechanics, Poznan University of Technology
ul. Jana Pawła II 24, 60-965 Poznań, Poland
roman.starosta@put.poznan.pl

Abstract

The paper presents a new method of vortex core detection developed for use in CFD simulation result analysis. Apart from the conventional approach involving vector algebra, mainly the Lambda2 method, it focuses on the identification of certain features in a graphic representation of the velocity field. It is done by generating a series of slices of the said field in the postprocessing software and training a Convolutional Neural Network (AI) to recognize vortex cores. The neural network can be integrated into a simple python program and used to quickly identify vortex cores on a large number of images and translate their locations to coordinates of a CFD model for visualisation.

Keywords: Vortex core tracking, Lambda 2, AI, Neural Network

1. Introduction

The accurate and precise description of the external flow over bodies plays a crucial role in many fields of engineering activities like aircrafts industry, shipbuilding, civil engineering, offshore structures, or turbomachinery. Aerodynamics is also a key element in successful vehicle design. The external flow has great importance in overall car efficiency and ride stability. Flow over the car is fully three-dimensional, turbulent, and unsteady. Besides, recirculation bubble formation and vortex detachment can be found in the area. These phenomena have a large impact on key factors in aerodynamic studies, lift and drag coefficients, relevant for driving stability and energy efficiency, respectively. Identifying vortex formation locations gives the possibility of lowering drag and reducing noise generated by the flow. The in-depth study of the turbulent flow around two basic car geometries (Ahmed and Asmo) is presented by Aljure et al [1].

Many papers are dealing with the problem of vortex shedding and many vortex identification methods have been proposed [2,3,4]. However, a clear classification of the methods (together with application areas) is still not available in the literature. As a result, it is very difficult for researchers to choose the right method for testing turbulent flow. The quantitative comparisons between different methods and their review in great detail, with many illustrating examples, are presented by Zhang et al [5].The various types of vortex generation and the related response characteristics of bluff bodies are described in [6]. There are presented many experiments illustrating and explaining vortex shedding phenomena.

Over the past two decades Computational Fluid Dynamics (CFD) became an important tool in the industry and research as the great increase in computational power of an average CPU computational made it possible to conduct analyses on a typical personal computer. Due to the nature of fluid flows, most practical applications relate to turbulent motion. Modeling of this phenomenon is crucial within CFD, however efficient recognition and tracking of vortices still pose a challenge.

2. Classical CFD approach

As the vortices are simply regions in a fluid revolving around an axis line the logical approach to this problem is the mathematical analysis of the velocity field.

One of the widely used algorithms called Lambda2 vortex criterion was proposed by Jeong and Hussain [2]. Applying it to a chosen point in the fluid determines whether this point is a part of a vortex core. The first step of the method is calculating the gradient of the velocity vector $\boldsymbol{J}$ from the velocity field acquired from CFD simulation

$$\boldsymbol{J} \equiv \nabla\vec{u} = \begin{bmatrix} \partial_x u_x & \partial_y u_x & \partial_z u_x \\ \partial_x u_y & \partial_y u_y & \partial_z u_y \\ \partial_x u_z & \partial_y u_z & \partial_z u_z \end{bmatrix}, \tag{1}$$

where $\vec{u}$ is the velocity vector and u_x, u_y, and u_z are velocity vector components.
It is then decomposed into its symmetric and antisymmetric parts **S** and **Ω**

$$\boldsymbol{S} = \frac{\boldsymbol{J}+\boldsymbol{J}^T}{2}, \quad \boldsymbol{\Omega} = \frac{\boldsymbol{J}-\boldsymbol{J}^T}{2} \tag{2}$$

.

The three eigenvalues of $\boldsymbol{S}^2 + \boldsymbol{\Omega}^2$are calculated for every point and ordered in such a way that $\lambda_1 \geq \lambda_2 \geq \lambda_3$. If for a given point $\lambda_2 < 0$, the point is a part of a vortex core, hence the method's name. The whole point of this mathematical operations is to neglect the effects of viscosity and non-stationarity and then look for the area of the lowest pressure as it achieves the lowest value in the vortex core.

This method is implemented in many CFD software packages and the Lambda2 coefficient can be easily acquired from the simulation results. It has, however, few drawbacks: distinguishing individual vortices can be difficult as they are usually surrounded by a large number of smaller vortices, graphic representation on static images can be ambiguous (Fig. 1) and, as Lambda2 is calculated for every point, it significantly increases the size of a result file.

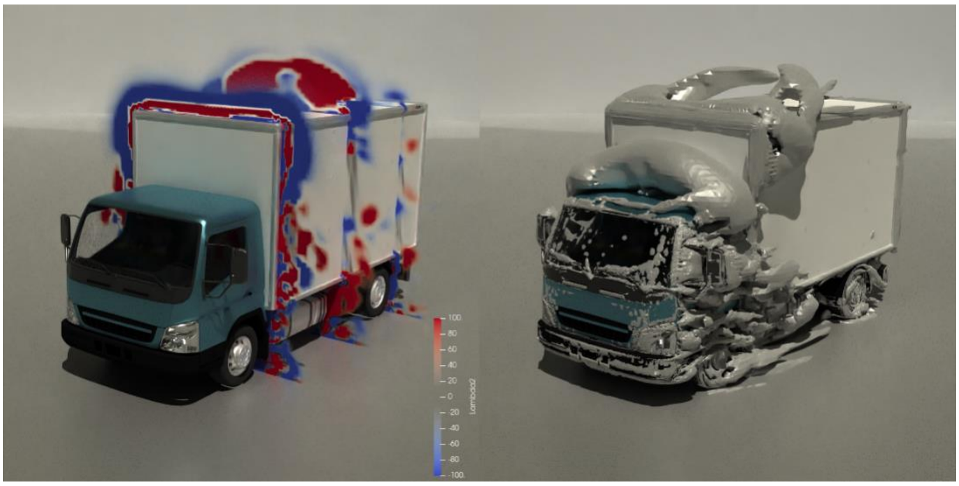

Figure 1. Visualisation of the Lambda2 coefficient

3. AI-based method

Recent developments in the field of computer science, mainly the advent of convolutional neural networks (CNN), gives us the possibility to approach the problem from a different angle.

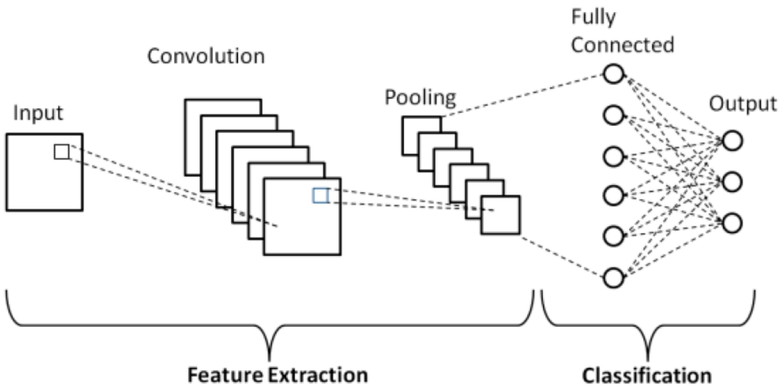

Figure 2. Typical Convolutional Neural Network (figure by Phung et al.,[1])

CNNs work by applying a series of mathematical operations called convolutions to an image and by supplying their output to a classical Neural Network (Fig. 2). CNN is trained on a large number of images and through the process of machine learning it gains the ability to associate certain features of an image with the type of the object presented in it. Using the CNNs one can quickly find certain features in a given image.

The approach described in this paper is based on training a CNN on a series of slices of velocity field generated using the line integral convolution (LIC) vector field visualization technique (Fig. 3). Slices were set 10 mm apart along the X-axis (flow direction) – 1000 images over the distance of 10 meters.

Figure 3. Visualization of the velocity vector field (LIC)

Two hundred and fifty vortex cores were manually identified on 25 of the images generated from the results of another CFD simulation. A simple python program was created for this task. Locations of the vortex cores are selected by clicking on the image and saved in json file which is accepted by neural network software.

The convolutional neural network chosen for this task was Keras-RetinaNet [7], based on ResNet50 architecture – predefined set of 50 convolutional layers (Fig. 5). It used so-called pre-trained weights – a large majority of the training process was performed on more powerful computers on a large set of random images representing many types of objects.

Figure 4. Vortex cores located by the CNN.

Keras is a library that provides a Python interface to artificial neural networks. The machine learning library used here is TensorFlow which was developed by Google for their internal use and later released as an open source software.

The computer used for this purpose is an ordinary personal desktop PC with an Ubuntu operating system. Training took approximately 3 hours. Script used to load images and generate sets of images for training and testing was written in python. All the software used is free and open-source.

The obtained model was then used to predict the locations of vortex cores on the whole series of 1000 images, predicted positions being saved to another json file. Simple script written in python was used to translate locations of vortex cores on images to the coordinate system of CFD model.

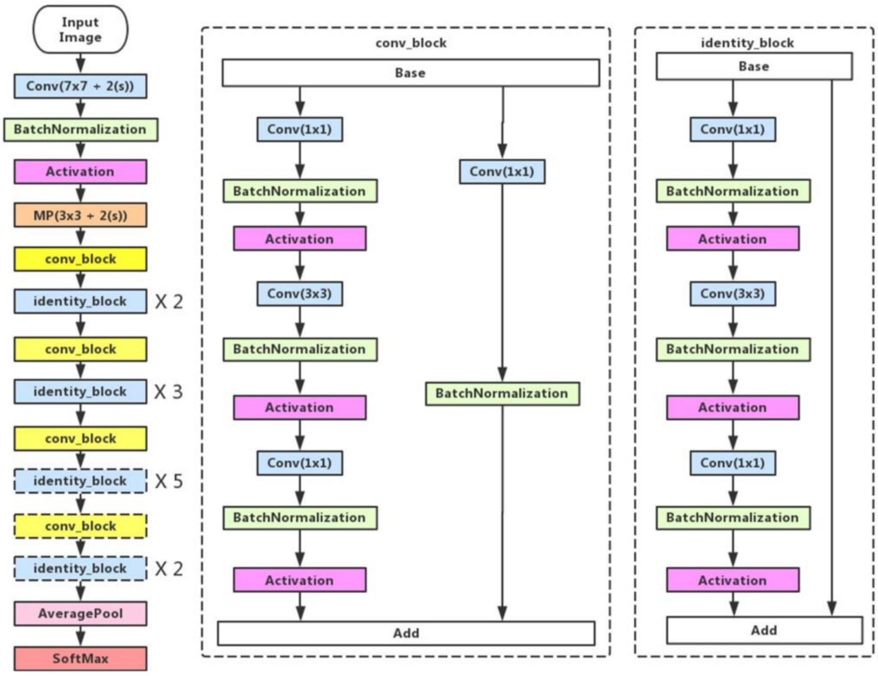

Figure 5. Block diagram of a ResNet50 architecture (figure by Ji et al.[8])

Points obtained in this way were saved in vtu file (VTK) for visualization in Paraview (Fig. 4).The predicted locations correlated with the velocity field vectors are shown in Fig. 6. The same script was used to remove a few false positives and to connect the remaining points into the vortex core center lines (Fig. 7).

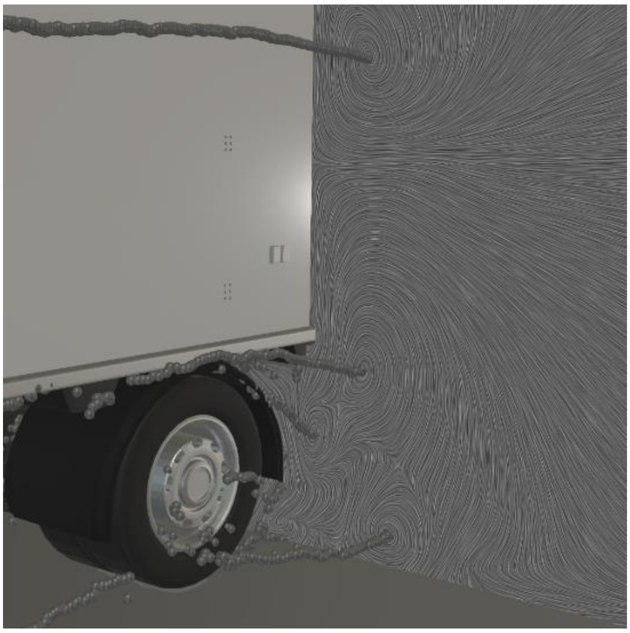

Figure 6. Vortex cores located by the CNN in relation to velocity field vectors

Figure 7. Vortex core lines

4. CFD simulation

Description of both Lambda2 method and proposed AI-based approach are based on a CFD simulation performed for the purpose of this paper. OpenFOAM - a free and open-source CFD toolbox was used. It is developed by OpenCFD Ltd company and used both commercially and academically.
Particular solver used here is called simpleFoam. It is a steady-state solver for incompressible, turbulent flows. CFD model was generated in the OpenFOAM's internal meshing utility. A 3D geometry of a light-duty commercial vehicle was downloaded from the internet and modified for this purpose using Blender - a free 3D computer graphics software. Geometry of a vehicle was placed in the middle of a CFD domain in a shape of a vertical cylinder 12 meters high and 56 meters in diameter (Fig. 8).

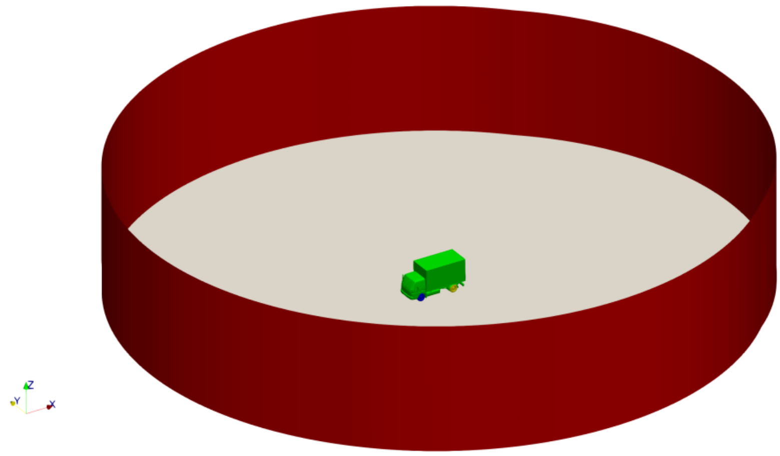

Figure 8. CFD domain

Mesh used in the simulation consists of 18 million polyhedral elements, their size ranging from 20 to 500 mm (Fig. 9). Flow velocity was set to 72 kph. The sides and top of a domain use the freestream velocity boundary condition, the bottom uses a moving wall condition, with the movement velocity the same as the flow velocity. The wheels are rotating walls, their rotational velocity matching the linear motion of the bottom wall. Turbulence model was set to k-omega SST [9], typical for this type of simulation, with turbulence kinetic energy value on the inlet equal 0.24 J/kg. It took approximately 14 hours to generate mesh and conduct the simulation on an ordinary desktop computer.

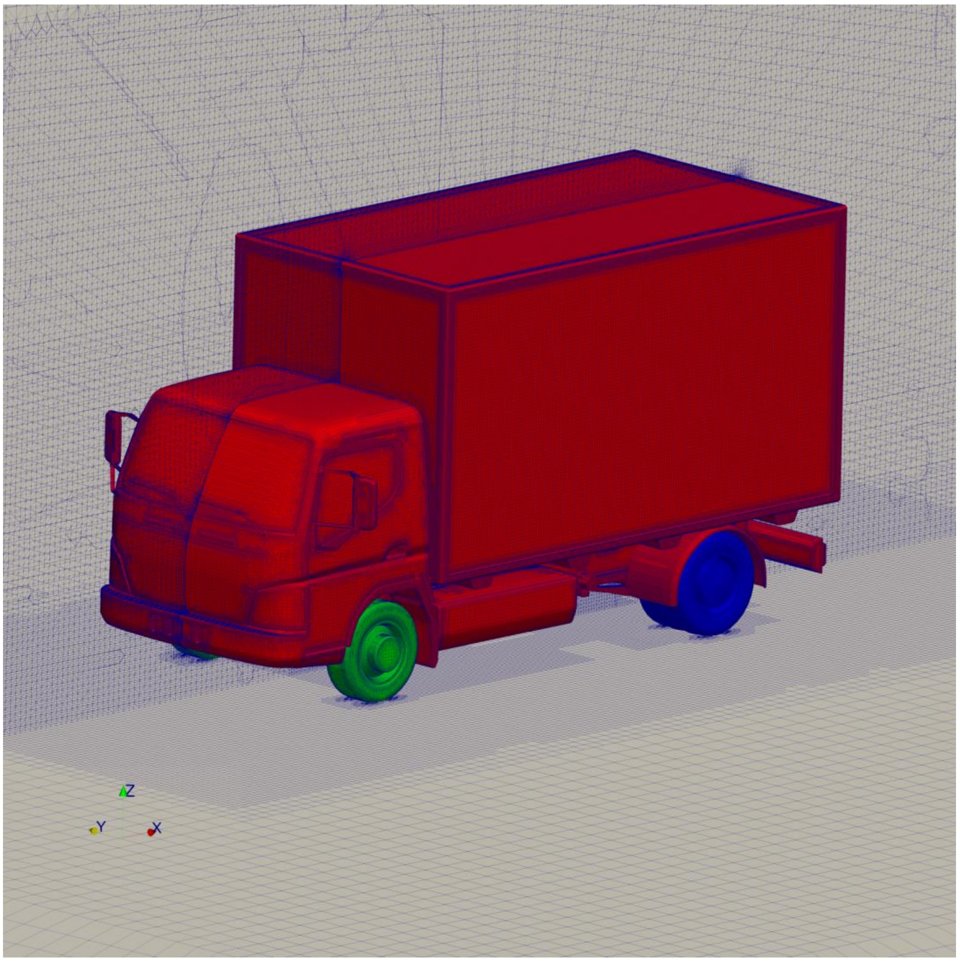

Figure 9. CFD mesh in the vicinity of vehicle

5. Conclusions

The method described in the paper gives very good results in a short time and does not require additional data beyond the velocity field which is very important in time-consuming and data-heavy CFD calculations. When using cloud computing services the additional disk space and data transfer volume directly contributes to the cost of simulation.

Moreover, as the Lambda2 values are calculated on the points of the CFD mesh the accuracy of the localization of vortex centres is affected by the size of the mesh while the LIC vector field visualization technique and, consequently, the AI-based method of vortex core tracking is independent of the mesh point location and thus largely

unaffected by mesh size. This allows for using larger mesh sizes and shorter computation times without the loss of results quality.
As the appearance of vortices is quite characteristic the proposed approach can be easily applied to different types of CFD simulations without the need for retraining of the Neural Network. By using simple script the process of vortex core tracking can be performed automatically after the CFD simulation ends. It is also possible to add this method to a visualization software in the form of a plugin.

The method can be further improved by training on larger sets of data and larger images, tracking vortices along other axes, and by distinguishing the clockwise and counter-clockwise vortices. Post-processing of the obtained results can also be improved by i.e. smoothing of the vortex core lines, better algorithm for false-positive removal, patching the small gaps in core lines.

Acknowledgments

This work was supported by the grant of the Ministry of Science and Higher Education in Poland, 02/21/SBAD/3558.

References

1. D. Aljure, I. Rodriguez, O. Lehmkuhl, R.Borrell, A.Oliva, *Flow and turbulent structures around simplified cars models*, Conference Paper: Conference on Modelling Fluid Flow (CMFF'12), Budapest, Hungary, September 4-7, 2012.
2. J. Jeong and F. Hussain. *On the Identification of a Vortex*. J. Fluid Mechanics, 285 (1995) 69 – 94.
3. Z.J. Taylor, E. Palombi, R. Gurka, G.A. Kopp, *Features of the turbulent flow around symmetric elongated bluff bodies*, J. Fluid Struct., 27(2) (2011) 250 – 265.
4. Z. Khatir, *A boundary element method for the numerical investigation of near-wall fluid flow with vortex method simulation*, Eng. Anal. Bound. Elem., 28 (2004) 1405–1416.
5. Y. Zhang, K. Liu, H. Xian, X. Du, *A review of methods for vortex identification in hydroturbines*, Renew.Sust.Energ. Rev., 81 (2018) 1269-1285
6. M. Matsumoto, *Vortex shedding of bluff bodies: a review*, J. Fluid Struct., 13 (1999) 791 – 811.
7. TY Lin, P. Goyal, R. Girshick, K. He, P. Dollar, *Focal Loss for Dense Object Detection*, arXiv.org 1708.02002 (2017).
8. Q. Ji, J. Huang, W. He, Y. Sun, *Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images*, Algorithms (2019) 12(3) 51.
9. F.R. Menter, *Two-equation Eddy-Viscosity Turbulence Models for Engineering Applications*, AIAA Journal. 32(8) (1994) 1598 – 1605.
10. V.H. Phung, E.J. Rhee, *A Deep Learning Approach for Classification of Cloud Image Patches on Small Datasets*, Journal of information and communication convergence engineering, 16(3) (2018) 173 – 178.