3D Flow Field Estimation around a Vehicle Using Convolutional Neural Networks

Fangge Chen
fanggechen@mail.nissan.co.jp

Nissan Research Center, Mobility & AI Laboratory
Nissan Motor Co., LTD.
Kanagawa, Japan

Kei Akasaka
kei-akasaka@mail.nissan.co.jp

Integrated CAE and PLM Department, Aerodynamics CAE Group
Nissan Motor Co., LTD.
Kanagawa, Japan

Abstract

Flow fields, including velocity and pressure fields, are typically used as references for vehicle shape design in the automotive industry to ensure steering stability and energy conservation. Generally, flow fields are calculated using computational fluid dynamics (CFD) simulations which is time-consuming and expensive. Therefore, a more efficient and interactive method is desired by designers for advanced shape discussion and design. To this end, we propose a fast estimation model using 3D convolutional neural networks. We employ a style extractor to obtain sufficient deep features of each vehicle shape and apply them using adaptive instance normalisation to improve the estimation performance. In addition, a proposed loss function which mainly includes a slice-weighted loss function is used to train the estimation model. The findings show that our proposed method outperforms previous studies, especially on flow field estimation in wake regions and regions near the vehicle surface. Therefore, the proposed method allows designing vehicle shapes while ensuring desirable aerodynamic performance within a much shorter period than extended CFD simulations.

1 Introduction

Flow fields of a vehicle mainly include velocity and pressure fields. In the automotive industry, they are advantageous for vehicle shape design in terms of ensuring desirable aerodynamic performance, which is directly associated with steering stability and energy conservation. For instance, to reduce the air resistance of a vehicle, designers often analyse the pressure distribution on the vehicle surface by referring to the pressure field. Generally, flow fields are calculated using computational fluid dynamics (CFD) simulations based on the Navier-Stokes equations [1] or the lattice Boltzmann method [2]. However, CFD simulations are time and cost intensive. An accurate calculation of CFD often takes several days. Most designers prefer discussing the aerodynamic performance of several design schemes in a one-hour meeting. An interactive and fast method is therefore needed to generate flow fields for evaluating aerodynamic performance.

Unlike flow field calculations using CFD simulations, recent studies have shown the feasibility of quickly estimating flow fields using machine learning methods. M. Mrosek *et al.* [3] proposed a method based on reduced-order modeling using the Kriging model which is similar to Gaussian process and Proper Orthogonal Decomposition plus Interpolation (POD+I) to predict flow fields of vehicle. However, the method can only work on the parameterized geometric features of a baseline vehicle shape and the pre-specific design space [4], which limits the discussion because designers always hope to change the arbitrary regions and discuss a several types of design schemes. N. Umetani *et al.* [5] proposed a method for encoding vehicle mesh data called PolyCube. Then, they used the encoded data to estimate the flow fields interactively using the Gaussian process. Raissi *et al.* [6] proposed a Navier-Stokes informed neural network. They employed automatic differentiation to calculate the derivatives of outputs. These derivatives were used to design a loss function and to train neural networks satisfying the Navier-Stokes equation. This consideration gradually grew into physics-informed neural networks (PINNs) in several studies [7, 8] which use physical laws as a loss function to train the estimation model. Current PINNs lack generalization for vehicle shape, and the training is difficult to converge. Nonetheless, these studies show the importance of designing loss functions for a better estimation performance.

Because convolutional neural networks (CNNs) have yielded successful performance in image recognition and object detection tasks, several studies [4, 9, 10, 11, 12] have demonstrated the potential of CNNs in the evaluation of aerodynamic performance. These studies proposed the use of CNNs to estimate dimensionless quantities, such as lift coefficients, to estimate 2D flow fields from a shape image or calculate 3D flow fields from simple geometric shape data. J. Musil *et al.* [13] proposed a U-net architecture [14] of residual 3D CNNs to estimate flow fields in real time when the architectural design was modified. The aforementioned studies have garnered a significant amount of interest with regard to the application of CNNs in contributing to aerodynamic analyses based on vehicle shape data.

In this study, we focus on estimating 3D flow fields, which include velocity and pressure fields, from 3D vehicle data. We generate a dataset for estimating flow fields from the unsigned distance function (uSDF) of a vehicle shape. Next, an estimation model based on 3D CNNs is proposed to implement our tasks, in contrast to previous studies designed for estimating 2D flow fields or 3D flow fields in a small space of simple 3D shapes such as a cube or cylinder. We apply adaptive instance normalisation to our estimation model to generate flow fields closely related to the vehicle shape. In addition, we proposed a novel loss function, mainly including a slice-weighted loss function, to train the estimation model by focusing on wake regions which is around the rear of vehicles and regions near the vehicle surface where the previous methods are difficult to estimate. The test results show that our proposed method outperforms the previous methods, especially on the flow field estimation in those target regions where the previous methods are difficult to estimate. Estimating flow fields from uSDF of a vehicle only need 0.3 seconds on Nvidia RTX2070 GPU, which makes designers discuss the aerodynamic performance of several design schemes in a short period.

2 Related Works

Compared with the estimation performed using a dimensionless quantity, such as lift or drag coefficients, the flow field considers information about the manner in which air passes through the object and how the force is applied to the object's surface. N. Umetani *et al.* [5] estimated the velocity field and pressure distribution on a vehicle surface using Gaussian process. A PolyCube method was proposed to reduce the dimensions of vehicle shape representations to develop an interactive method for implementation. The PolyCube method, however, cannot correspond to certain parts of the vehicle, such as floor parts or door mirrors, which considerably influence the aerodynamic performance. In this study, we use a simple data representation applied to all parts of a vehicle and show that the CNN model can also perform an efficient flow field estimation.

The task of flow field estimation can be easily associated with an image style transfer, such as pix2pix [15]. Inspired by the CNN architecture of an image style transfer, Guo *et al.* [10] proposed a CNN model to estimate the velocity field in real time based on 2D shape data and simple 3D shape data, which validated the efficacy of the proposed method. Two types of datasets containing vehicle side-view images were used to train the model, and average relative error rates of 9.04% and 16.53% were obtained for the datasets. Bhatnagar *et al.* [12] proposed a method similar to the one mentioned above to estimate the velocity and pressure fields of an aerofoil. They employed the gradient difference loss [16] to train the model and showed that the estimation results significantly reduced errors in the wake region of aerofoils. Recently, physics-informed neural networks (PINNs) [7] have been used with increasing frequency, which train neural networks using physical laws as a part of the loss function. O. Hennigh *et al.* [8] also proposed a method based on PINNs. However, they used signed distance functions (SDF) for loss weighting, which results in a convergence speed increase and also improved accuracy in some situations.

Although our proposed estimation model is also based on CNNs, we find that previous methods are weak in estimating 3D flow fields, especially in the wake region or the regions near the vehicle surface. Thus, in this study, we propose a novel estimation model to improve the performance on 3D cases. Moreover, from the aforementioned studies, we find that the loss function is related to the performance of estimation accuracy. In this study, we propose a slice-weighted loss function and use continuity equation from fluid mechanics to train the estimation model.

3 Proposed Method

In this section, we introduce our proposed flow field estimation method in detail. Because there is no available dataset, we generate a dataset that includes the uSDF of vehicle shapes and corresponding flow fields at first. Then, we introduce the proposed estimation model architecture whose inputs are the uSDFs and outputs are the 3D flow fields around the vehicle. An overview of the estimation model is presented in Fig.1. To improve the estimation near the surface or in the wake region, we introduce the proposed loss function and train the model with a physical rule.

3.1 Dataset

For the training estimation model, a dataset that includes the 3D shape information of vehicles and corresponding flow fields is urgently required. Umetani *et al.* [5] released a dataset. However, this dataset cannot be directly used in the present study because the vehicle data lacks metric information, and the calculation condition of flow fields is significantly different from that of our practice work. Therefore, we generated a dataset from scratch. Fig.2 shows the structure of our dataset, which includes a total of 1,121 cases of vehicles from different categories, such as sedans and SUVs. Each case in the dataset contains an uSDF

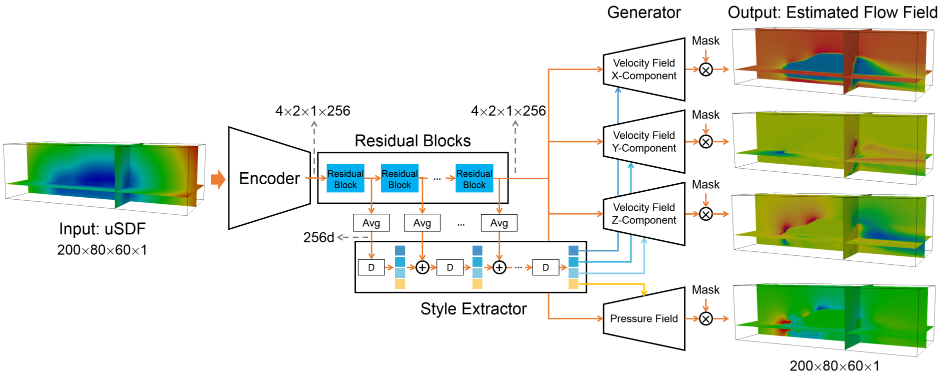

Figure 1: An overview of flow field estimation model architecture.

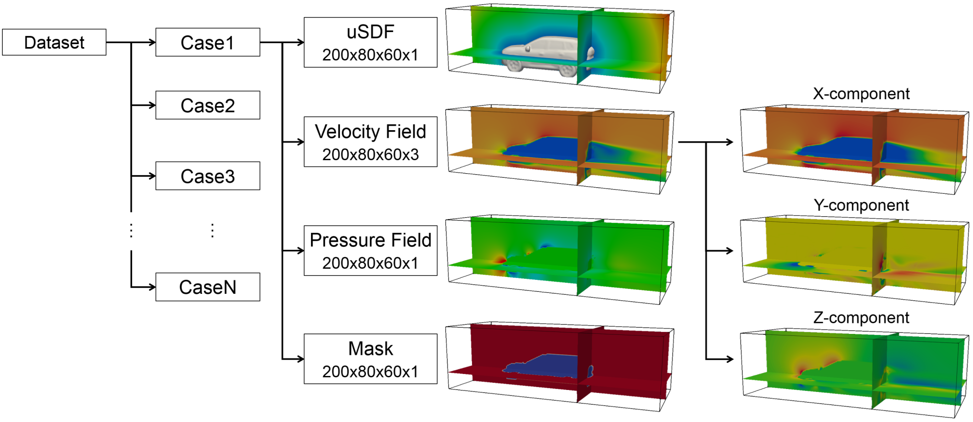

Figure 2: The structure of our dataset

of the vehicle shape as the input of the estimation model, a ground truth velocity field and pressure field as the output, and a binary mask showing the existing flow fields. We selected a $10m \times 4m \times 3m$ Cartesian space and placed the vehicle in the space. The uSDF of this space was calculated to represent the vehicle shape, and the corresponding flow fields in the same space were calculated using CFD software which sets the condition that vehicles drive in the air with 120 km/h speed. We sampled the aforementioned space with a 0.05m pitch so that the size of the uSDF is $200 \times 80 \times\ 60 \times 1$. The air velocity is a vector that includes three components along the axes; hence, the size of the velocity field is $200 \times 80 \times 60 \times 3$, whereas the size of the pressure field is $200 \times 80 \times 60 \times 1$ because the pressure is scalar. In addition, the binary mask of each case, where one means the region that allows air to pass through, was added into the dataset. These masks are very useful for calculating the loss that will be introduced later.

3.2 Architecture of the Estimation Model

We adopt an encoder-decoder architecture similar to that used in previous studies. To deal with more complex 3D data, we design an estimation model based on 3D CNNs. Although the estimation models in previous studies in [10] and [12] are simple, considering the capacity of the model for 3D flow field estimation, we need to increase the complexity of the model. The estimation model includes four parts: an encoder, several residual blocks, a style extractor and four generators of each flow field component.

The encoder is designed based on the VGGNet [17] structure. Considering that our input uSDF and output flow fields are $200 \times 80 \times 60$, we applies instance normalisation [18] because the training batch size cannot set a large value. The input should be rescaled from the uSDF data into the range of [0, 1] using the exponential function $exp(-\text{uSDF})$. The output feature of the encoder is $4 \times 2 \times 1 \times 256$. In contrast to previous studies, we avoid using a fully connected layer, which causes redundant calculations. Next, several residual blocks are used to extract deeper features. These blocks benefit from a residual neural network (ResNet) [19], such that the shortcut connection can help in training the estimation model well with more layers. The output feature of the final residual block is sent into four generators, which output three components of the velocity and pressure fields. All the generators have the same structure. Inspired by StyleGAN [20], we assume that it is possible to control generators using the style feature of a vehicle so that the generated flow fields can be in accordance with the vehicle shape features. Therefore, we design a style extractor to extract the style of each vehicle shape. The output of each residual block should be converted to a one-dimensional feature by calculating the mean of each channel. To keep the scale of each input one-dimensional feature, we normalise the feature before sending it into the dense layer or being added to the intermediate feature. The normalisation we use is based on the local response normalisation in [21] and [22], which makes all elements into a unit length. Because each component of the flow fields has four generators, the style features are also calculated separately.

We follow the methods outlined in [20] to apply adaptive instance normalisation (AdaIN) [23] to integrate the style feature into the generator. We use 3D deconvolution layers to gradually enlarge the size and applied 3D convolution layers to refine the generation. Similar to the encoder, we adopt instance normalisation in the generator. One generator will output one component of the flow field with a size of $200 \times 80 \times 60 \times 1$. Finally, the output multiplies the binary mask of the vehicle element-wise to make the value inside the vehicle and on the surface zero because air cannot reach these regions.

3.3 Loss Function

The loss function is shown in equation (1), where $Loss_{slice}$ is the slice-weighted L1 loss, $Loss_{gradient}$ is the gradient difference loss and $\operatorname{div}\mathbf{U}$ is the divergence of the velocity field.

$$Loss_{total} = Loss_{slice} + \lambda_1 Loss_{gradient} + \lambda_2 \operatorname{div}\mathbf{U} \tag{1}$$

Although L1/L2 loss is widely used for training CNNs, it is necessary to modify these basic loss functions. The value of the velocity and pressure fields should be zero on the surface or inside the vehicle. In these regions, the estimated results are equal to the ground truth data because of the element-wise multiplication with the mask. For this reason, calculating the loss by subtracting them directly leads to a small mean value of the total space. We hope to use the larger loss and calculate only in the region where air can pass through by using a mask. Therefore, we propose a slice-weighted loss function defined in equation (2), where $\tilde{x}$ is a estimated component of the flow field and x is the ground truth. The loss function regards $L \times W \times H$ (Here is $200 \times 80 \times 60$) 3D data as a series of images, such as being sliced from different viewpoints. We found that it was difficult to accurately estimate the flow fields near the vehicle surface and wake regions. In our case, the slices, including the boundary of the vehicle surface, are the regions where only small air passes. Calculating each loss in the slice and then obtaining the average where mask value equals one lead to an increase in the weight of loss, which is included in the flow field region of this slice. This means that the loss can be weighted by the slices, increasing the weight if slice includes the vehicle boundary. Because those target regions where we hope to improve estimation accuracy are always around boundary, the loss in the regions can be enhanced by the weighted method described in equation (2). Consequently, the proposed slice-weighted loss function makes estimation model training focus on improving estimation in both the regions near the vehicles and wake regions.

$$\begin{aligned} Loss_{slice}(x,\tilde{x},mask) = & \frac{1}{L}\sum_{i=0}^{L-1}\frac{\sum_{j=0}^{W-1}\sum_{k=0}^{H-1}(\tilde{x}_{i,j,k}-x_{i,j,k})}{\sum_{j=0}^{W-1}\sum_{k=0}^{H-1}mask_{i,j,k}}+ \\ & \frac{1}{W}\sum_{j=0}^{W-1}\frac{\sum_{i=0}^{L-1}\sum_{k=0}^{H-1}(\tilde{x}_{i,j,k}-x_{i,j,k})}{\sum_{i=0}^{L-1}\sum_{k=0}^{H-1}mask_{i,j,k}}+\frac{1}{H}\sum_{k=0}^{H-1}\frac{\sum_{i=0}^{L-1}\sum_{j=0}^{W-1}(\tilde{x}_{i,j,k}-x_{i,j,k})}{\sum_{i=0}^{L-1}\sum_{j=0}^{W-1}mask_{i,j,k}} \end{aligned} \tag{2}$$

Bhatnagar *et al.* [12] reported that using gradient difference loss, which calculates the average difference of spatial gradient between the ground truth and estimated flow fields, can significantly reduce the estimation error. Accordingly, this method is deemed applicable to our task in training a more accurate model.

Moreover, the flow around the vehicle can be regarded as an incompressible flow in which the density is constant. For this condition, the divergence of the velocity field should be zero and can be derived from the continuity equation. We add this physical constraint into the loss function to train the estimation model of the velocity field.

3.4 Training Conditions

We train the estimation model using four Adam optimisers for each component of the flow field. Each optimiser is applied to the feature extractor, residual block, corresponding style extractor and generator to update their parameters. The proposed loss function is used by setting λ_1 to 0.1 and λ_2 to 0.01 when training the estimation of velocity fields. λ_2 was set to 0 when training the estimation of the pressure field because the divergence of velocity cannot influence it. We set the batch size to 4 and train the estimation model for 1,000 epochs with the initial learning rate, which is set to 10^{-3}, and then exponentially decay to 10^{-4} during the first 500 epochs. We implement all the estimation models using TensorFlow 2.x and train them on one Nvidia Tesla V100 GPU with 32GB of GPU memory.

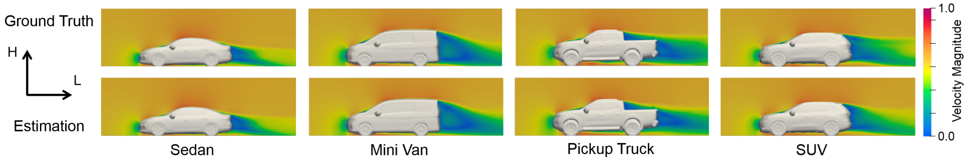

Figure 3: Estimated velocity field of several vehicles in test set

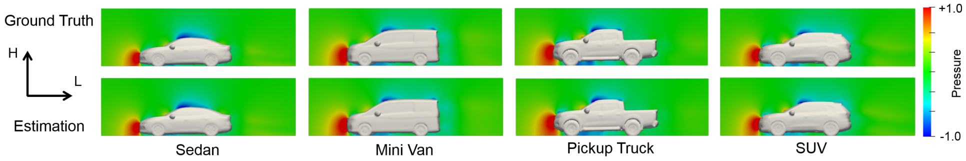

Figure 4: Estimated pressure field of several vehicles in test set

4 Evaluation

To evaluate the proposed method, we split the dataset into a training set and a test set, including 1,098 cases and 113 cases, respectively. Part of the test data were randomly selected, and the remaining cases included some special designs, which were selected by domain experts. Furthermore, we randomly split the training set into nine groups on average, each of which included 122 cases, which were employed for cross-validation. We used eight groups for the training model and the remaining one group for validation. All the evaluations were performed on a notebook computer which includes a Nvidia RTX2070 GPU.

4.1 Results of Flow Field Estimation

The estimated flow fields of several typical vehicle cases from the test set are shown in Fig. 3 and Fig. 4 [1]. We found that the proposed estimation model can be applied to different categories of vehicles. Compared to the ground truth data, the estimated flow fields were close to them. Velocity magnitude of both ground truth and estimated on $L \times H$ slice are shown in Fig. 3. It can be seen that the velocity field peculiar to vehicle, such as the stagnation area where the flow velocity is slow at the front of vehicle and the wake region with slow velocity around the rear of the vehicle, can be reproduced by the proposed estimation model. Fig. 4 shows the slices of pressure scalar on the same locations. The typical regions of pressure field, such as the high pressure at the front of the vehicle and the negative pressure on the top of the roof, can be reproduced by the proposed estimation model.

Types of MAE	Velocity Field (m/s)			Pressure Field (Pa)
	X-Component	Y-Component	Z-Component	
Average	0.494	0.252	0.230	3.738
Minimum	0.482	0.248	0.227	3.616
Maximum	0.509	0.256	0.234	3.897

Table 1: Mean absolute errors for each estimated component of flow fields in test set.

[1] Please note that we used the scaled unit-less ranges shown in the colorbars for velocity magnitude and pressure because of the data provider's confidentiality policy. However, the differences are allowed to show the actual range.

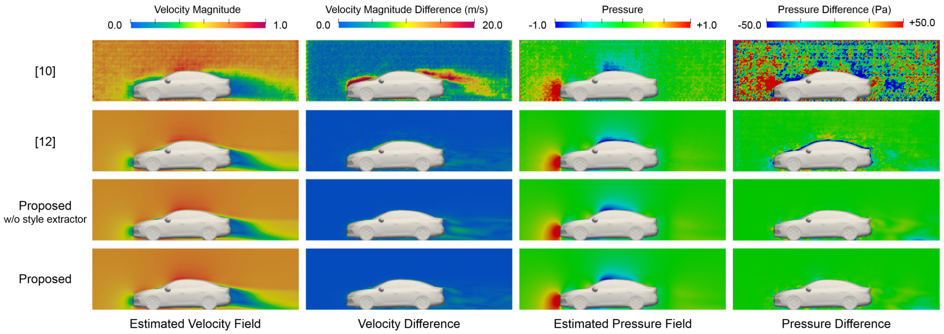

Figure 5: Estimation results and differences from ground truth using different methods.

Methods	Velocity Field (m/s)			Pressure Field (Pa)
	X-Component	Y-Component	Z-Component	
[10]	2.842	1.330	1.027	48.578
[12]	0.653	0.415	0.340	9.236
Proposed w/o style extractor	0.501	0.255	0.230	3.673
Proposed	**0.482**	**0.248**	**0.229**	**3.616**

Table 2: Comparisons against previous studies.

To evaluate the accuracy of the model, we calculated the mean absolute error (MAE) of the estimated velocity field and pressure field on the test set, in contrast to their ground truth, for each cross-validation. Table 1 shows the average, minimum and maximum MAE results of nine runs on the test set using different trained models with cross-validation. The average MAEs of the three components of the velocity field are 0.495, 0.252 and 0.230 m/s, and the MAE of pressure field is 3.758 Pa. In addition, the minimum and maximum MAEs are shown in Table 1, we found that the MAEs do not change significantly using different trained models. These results show the effectiveness of the proposed method which is able to provide a robust estimation performance. Likewise, these results show the reasons why a significant difference was not observed between the estimated flow fields and ground truth.

Using the proposed method, we can obtain the results of estimated flow fields from mesh data within 20 seconds, which include 10 $\sim$ 15 seconds for converting a vehicle mesh to uSDF data and 0.3 seconds for estimating the flow fields. The method can accelerate the vehicle shape design discussion with aerodynamic performance because the current process in practice usually requires at least one day. It becomes possible for designers to interactively revise the vehicle shape and know a rough aerodynamic performance in a short period.

4.2 Comparison

For comparison with previous studies, we faithfully reimplemented two estimation models based on [10] and [12]. However, because these models were designed mainly for images, we revised them to fit our data size and to support the training of 3D data. The details of these estimation models are provided in the supplementary material. We trained these models using the first group of training data as a validation set and the remaining training data as a training set.

Table 2 shows the MAEs of each estimation model running on test set. The results ob-

Loss Functions	Velocity Field (m/s)			Pressure Field (Pa)
	X-Component	Y-Component	Z-Component	
L1 Loss	0.882	0.261	0.251	4.887
L2 Loss	1.376	0.869	1.016	9.701
SDF Weighted L1 Loss [8]	1.125	0.304	0.347	6.745
Exponential SDF Weighted L1 Loss	1.355	0.267	0.290	7.808
L2 Slice-weighted Loss	0.651	0.506	0.469	5.361
Proposed (w/o velocity divergence)	0.499	0.255	0.230	3.877
Proposed	**0.482**	**0.248**	**0.229**	**3.616**

Table 3: Mean absolute errors for each estimated component of flow fields on test set with training by different loss functions.

viously show that our proposed estimation model is able to provide a more accurate flow field estimation. Fig. 5 shows the flow fields of a sedan case in the test set estimated by these models. The differences between the estimated flow field and ground truth are also shown in this figure. As mentioned before, these models were designed for estimating the 2D flow fields or the 3D flow fields of simple 3D shapes. It is difficult to apply them to our task by just revising to fit our data size. The first row in Fig. 5 shows that the estimation model does not have enough capability to estimate 3D flow fields. Although the outputs in the second row look more like flow fields, the differences from the ground truth show that the large errors occur in the regions near the vehicle surface and wake region. Comparing with these previous methods, our proposed estimation model is more suitable for estimating the flow fields in these regions benefiting from the architecture which extracts the style features of vehicle shape from residual blocks and applies them to control generation of flow fields. Table 2 also shows the results of the estimation model without style extractor module, which verify the effectiveness of the proposed estimation model architecture. Furthermore, our proposed loss function makes the model concentrate on learning the flow fields in those target regions which is difficult to estimate. This also leads to a more accurate estimation. We will introduce the effectiveness of the loss function in the next section.

4.3 Ablation Study of Loss Function

We consider that the proposed loss function plays an important role in training a more accurate estimation model. To verify its effectiveness, we trained the estimation model with different loss functions, including L1/L2 loss functions, and the slice-weighted L2 function. We also implemented the losses both described in [8] using SDF and inversely using the exponential function of SDF by $e^{-|SDF|}$ to weigh the L1 loss function. We replaced the $Loss_{slice}$ in equation (1) with the aforementioned loss functions respectively and kept the remaining components. Moreover, the estimation model without velocity divergence was trained to verify the effectiveness of this component in loss function. We split training data into training set and validation set, which is the same as the previous section, to train the models.

Fig. 6 shows the estimated velocity fields of the same sedan vehicle used in the previous section and the differences from the ground truth using the estimation model which was trained on different loss functions. We found that all the estimated velocity fields are close to the ground truth. However, the differences in Fig. 6 show that using L1/L2 loss functions or using SDF-weighted loss functions can produce the larger difference in the region near the vehicle surface or wake region. Training estimation model by slice-weighted loss makes the estimation with the less error. The MAEs of the each component of flow fields on the test set are shown in Table 3. The table clearly shows that using slice-weighted loss can improve the

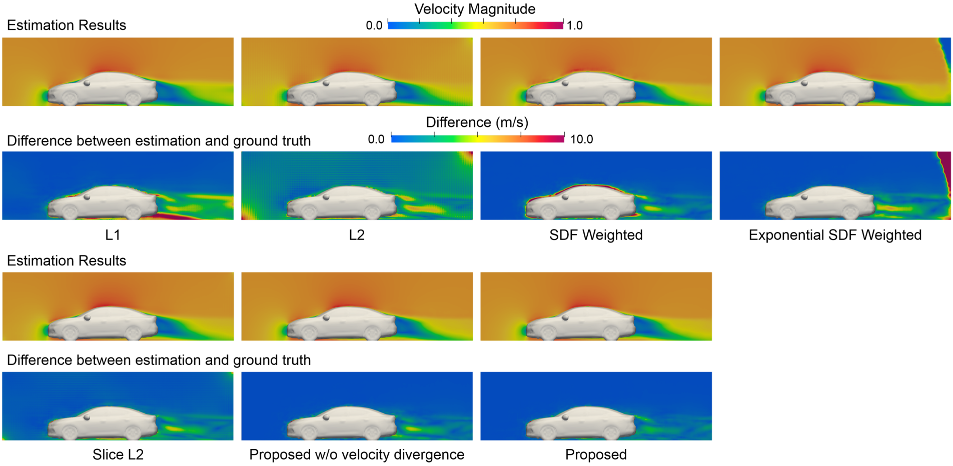

Figure 6: Estimation results by the proposed models trained with different loss functions and the differences from ground truth.

estimation accuracy. As explained in section 3.3, the proposed slice loss can obtain the larger loss to train the estimation model focusing on those target region. The table also shows that using velocity divergence as a part of loss function can contribute to estimation accuracy. Moreover, the result shows the reasons why we chose L1 loss rather than L2 loss: the L2 loss always decreases very early, making training a more accurate estimation model difficult.

5 Conclusions

In this paper, we proposed a 3D flow field estimation model based on 3D CNNs, which estimate the velocity and pressure fields around a vehicle in a 3D space from uSDF vehicle shape data. We employed a style extractor to obtain sufficient deep features of vehicle and applied them through AdaIN to improve the estimation. We trained the estimation model with a proposed slice-weighted loss function. The estimated results were very close to the ground truth, and a significant difference is difficult to be observed. We showed that the proposed estimation method outperformed previous methods and verified effectiveness of the proposed loss function by comparing it with other normal loss functions. This method makes it possible to discuss various vehicle design schemes with aerodynamic performance analysis in a short period.

6 Acknowledgements

We thank Takuya Nanri (Nissan Research Center Mobility & AI Laboratory, Nissan Motor Co., LTD.) for very useful discussions and suggestions.

References

[1] Yue-Hong Qian, Dominique d'Humières, and Pierre Lallemand. Lattice bgk models for navier-stokes equation. *EPL (Europhysics Letters)*, 17(6):479, 1992.

[2] Guy R McNamara and Gianluigi Zanetti. Use of the boltzmann equation to simulate lattice-gas automata. *Physical review letters*, 61(20):2332, 1988.

[3] Markus Mrosek, Carsten Othmer, and Rolf Radespiel. Reduced-order modeling of vehicle aerodynamics via proper orthogonal decomposition. *SAE International Journal of Passenger Cars-Mechanical Systems*, 12(3):225–237, 2019.

[4] Sam Jacob Jacob, Markus Mrosek, Carsten Othmer, and Harald Köstler. Deep learning for real-time aerodynamic evaluations of arbitrary vehicle shapes. *arXiv preprint arXiv:2108.05798*, 2021.

[5] Nobuyuki Umetani and Bernd Bickel. Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics (TOG)*, 37(4):1–10, 2018.

[6] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: A navier-stokes informed deep learning framework for assimilating flow visualization data. *arXiv preprint arXiv:1808.04327*, 2018.

[7] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[8] Oliver Hennigh, Susheela Narasimhan, Mohammad Amin Nabian, Akshay Subramaniam, Kaustubh Tangsali, Zhiwei Fang, Max Rietmann, Wonmin Byeon, and Sanjay Choudhry. Nvidia simnet™: An ai-accelerated multi-physics simulation framework. In *International Conference on Computational Science*, pages 447–461, 2021.

[9] Yao Zhang, Woong Je Sung, and Dimitri N Mavris. Application of convolutional neural network to predict airfoil lift coefficient. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1903, 2018.

[10] Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 481–490, 2016.

[11] Nils Thuerey, Konstantin Weißenow, Lukas Prantl, and Xiangyu Hu. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 58(1):25–36, 2020.

[12] Saakaar Bhatnagar, Yaser Afshar, Shaowu Pan, Karthik Duraisamy, and Shailendra Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64(2):525–545, 2019.

[13] Josef Musil, Jakub Knir, Athanasios Vitsas, and Irene Gallou. Towards sustainable architecture: 3d convolutional neural networks for computational fluid dynamics simulation and reverse designworkflow. *arXiv preprint arXiv:1912.02125*, 2019.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[16] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[18] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[23] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.