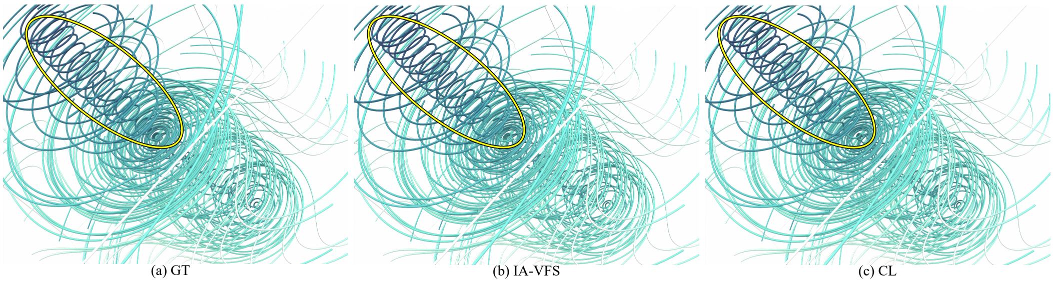
Integration-Aware Vector Field Super Resolution

S. Sahoo & M. Berger
Vanderbilt University



(a) GT (b) IA-VFS (c) CL

Figure 1: *Comparison of streamline results between our technique (b), ground truth (a) and the baseline (c) for tornado dataset, differences highlighted in yellow.*

Abstract

In this work we propose an integration-aware super-resolution approach for 3D vector fields. Recent work in flow field super-resolution has achieved remarkable success using deep learning approaches. However, existing approaches fail to account for how vector fields are used in practice, once an upsampled vector field is obtained. Specifically, a cornerstone of flow visualization is the visual analysis of streamlines, or integral curves of the vector field. To this end, we study how to incorporate streamlines as part of super-resolution in a deep learning context, such that upsampled vector fields are optimized to produce streamlines that resemble the ground truth upon integration. We consider common factors of integration as part of our approach – seeding, streamline length – and how these factors impact the resulting upsampled vector field. To demonstrate the effectiveness of our approach, we evaluate our model both quantitatively and qualitatively on different flow field datasets and compare our method against state of the art techniques.

CCS Concepts

• Computing methodologies → Neural networks; Reconstruction; • Human-centered computing → Scientific visualization;

1. Introduction

In the field of scientific computing, scientists generate high resolution flow-field data from large-scale numerical simulations for analysis and visualization purposes. High resolution vector fields provide highly-resolved structural details that enable complex analyses of flow behavior. However, there remain challenges in the workflow from data generation to post-hoc analyses, namely, bottleneck in file I/O in terms of space and network bandwidth for file transfer. Thus, scientists end up either simplifying the flow data or visualizing only part of it at a time, leading to incomplete analyses.

To mitigate this problem, in recent years vector field super resolution has evolved as a mature technique. Vector field super resolution refers to the task of recovering high resolution vector fields from their low resolution counterparts. Traditionally, interpolation techniques such as tri-linear interpolation or tri-cubic interpolation have been used for the purpose of super-resolution. However, these techniques lack the ability to capture the global flow behaviour since up-sampling is based on local information only. In recent years, deep learning based techniques have received significant attention in the literature and have shown to outperform these more traditional techniques. Yet a limitation common to existing methods is that no consideration is made to how upsampled vector fields are used in practice. Specifically, as part of a scientist's visual analysis workflow, it is extremely common to integrate the vector field to obtain streamlines, in order to discover more global, structural flow features. On the other hand, a key advantage to optimization-based techniques for super-resolution, e.g. deep learning methods, is that we may optimize for what we ultimately visualize, e.g. streamlines.

To this end, in this paper we propose an integration-aware approach to vector field super-resolution. Streamlines have a long history within visualization [SBGC20] , thus when considering how to use streamlines for super-resolution, there are a number of fac-

DOI: 10.2312/evs.20211054

tors to consider, ranging from the seeding technique to the length of integration. We show how to augment more traditional super-resolution objectives with integration-aware losses, and study how these factors impact the resulting optimization. Through experimentation on a number of datasets, we quantitatively and qualitatively show that it is possible to guide super-resolution to better reflect streamlines in ground-truth flow fields. Our contributions can be summarised as follows: (1) We introduce integration-aware optimization for vector field spatial super resolution using deep learning models. (2) We study the impact of factors common to streamlines as part of optimization. (3) We evaluate our approach on a range of vector field datasets and demonstrate superior results compared to existing methods, both quantitatively and qualitatively.

2. Related Work

Super resolution methods have received significant attention in the literature. Conventional methods like interpolation-based [Duc79; Key81] and reconstruction-based techniques [MO08; SXS08; DHX*09; YXYN15] are simple but fail to generate perceptually accurate images. Dong et. al [DLHT15] introduced a deep learning based pre-upsampling framework(SRCNN) outperforming more traditional approaches. Other techniques akin to SRCNN with varying network architecture and learning strategies [KKM16; SCI18] have been studied. Post-upsampling networks [DLT16; SCH*16] attempt to overcome the computational expense of pre-sampling networks. Following its success, most recent work uses this framework, varying the network architecture [LSK*17; TYL17; TLLG17; HCL*18; ZTK*18; LHAY17]. Many loss functions like content loss [JAF16], texture loss [SSH17] and adversarial loss [LTH*17] have also been explored to generate photo realistic image.

Deep learning for scientific visualization has gained significant attention in the recent years. Techniques have been developed for modeling volume rendering [BLL18; HWG*19], visualizing complex volumetric structures [CCJ*18], and viewpoint estimation within volume visualization [ST19]. Zhou et al. [ZHW*17] introduced volume upscaling using a CNN based model, while Xie et al. [XFCT18] proposed tempoGAN for temporally coherent super-resolution of volumetric flow fields. Han et al. [HW19] developed a generative network for temporal super-resolution of time varying volumetric data. Han et al. [HTW18] introduced FlowNet for interactive streamline clustering and selection using deep learning. Han et al. [HTZ*19] proposed a two-stage deep learning framework for high-quality flow field reconstruction using streamlines. Our work is related to the vector field super-resolution work of Guo et. al [GYH*20] where they optimize for the magnitude and angle of target vectors. We extend this method to target the optimization of integral curves in upsampled vector fields.

3. Methodology

3.1. Overview

In this work, our goal is to estimate a high-resolution vector field, denoted $\mathbf{V_h} \in \mathbb{R}^{w\times h\times d\times 3}$ of spatial resolution $(w \times h \times d)$, given its corresponding low-resolution counterpart, denoted $\mathbf{V_l} \in \mathbb{R}^{w'\times h'\times d'\times 3}$. We assume a subsampling factor r, a positive integer such that $w = r \cdot w'$, $h = r \cdot h'$, and $d = r \cdot d'$. Our approach to super-resolution is to learn a mapping that we denote as $f : \mathbb{R}^{w'\times h'\times d'\times 3} \to \mathbb{R}^{w\times h\times d\times 3}$, where f is parameterized as a volumetric convolutional neural network, following prior work [GYH*20]. We depart from Guo et al. in what we optimize: we would like the mapping f to upsample vector fields in such a manner that integral curves of $\mathbf{V_h}$ are preserved. Namely, integrating $f(\mathbf{V_l})$ produces streamlines that are as close as possible to streamlines of $\mathbf{V_h}$.

3.2. Integration-Aware Upsampling Loss

Loss functions used in super-resolution tend to focus on content given in the high-resolution target, e.g. pixels in a high-resolution image, or in our case, vectors in a high-resolution vector field [GYH*20]. This can be expressed as follows:

$$\mathcal{L}_M = \frac{1}{|\mathcal{P}|}\sum_{\mathbf{p}\in\mathcal{P}} \|f(\mathbf{V_l}))[\mathbf{p}] - \mathbf{V_h}[\mathbf{p}]\|_2\,, \tag{1}$$

where $\mathcal{P}$ indexes over vertices of a $(w \times h \times d)$ grid.

To ensure that our network can preserve streamlines, we introduce a loss function that is based on integral curves of, both, $\mathbf{V_h}$ and upsampled vector field $f(\mathbf{V_l})$, please see Fig. 2. Specifically, we denote $S = \{s_1, s_2, \ldots, s_n\}$ as a set of integral curves derived from $\mathbf{V_h}$, where $s_i = (\mathbf{p}_{i,1}, \mathbf{p}_{i,2}, \ldots, \mathbf{p}_{i,m})$ is a set of points on curve s_i. Given the upsampled vector field $f(\mathbf{V_l})$, we also form integral curves, taking the seed points from ground truth for integration. Specifically for streamline s_i, we integrate $f(\mathbf{V_l})$ starting at position $\mathbf{p}_{i,1}$ to obtain streamline $s'_i = (\mathbf{p}'_{i,1}, \mathbf{p}'_{i,2}, \ldots, \mathbf{p}'_{i,m})$, where $\mathbf{p}'_{i,1} = \mathbf{p}_{i,1}$. Our loss function is designed to ensure that the two curves remain close at *all* integrated positions:

$$\mathcal{L}_S = \frac{1}{n\times m}\sum_{i=1}^{n}\sum_{j=1}^{m} \left\|\mathbf{p}_{i,j} - \mathbf{p}'_{i,j}\right\|_2\,. \tag{2}$$

In practice, we combine the two loss terms, ensuring a balance between vector content ($\mathcal{L}_M$) and flow structure ($\mathcal{L}_S$):

$$\mathcal{L}_T = \lambda\mathcal{L}_S + (1-\lambda)\mathcal{L}_M \tag{3}$$

where λ is a hyper-parameter which determines the relative importance of integration-based loss – a high λ places large importance on integration. We optimize the loss via stochastic gradient descent, which requires backpropagating over the integration method of choice. However, integration schemes like Euler integration and Runge-Kutta, can be expressed as a differentiable function with respect to the vector field, assuming a differentiable form of interpolation for accessing vectors at arbitrary positions. In practice, we use trilinear interpolation, thus we may optimize the loss function $\mathcal{L}_T$ end-to-end.

What remains is a way to form the streamline set $\mathcal{P}$. We would like to ensure the streamlines are representative of predominant flow features. To this end, we consider **seeding** and **integration length**, studied further in Sec. 4.3.

Seeding: The starting positions from which to integrate are important for ensuring flow features are preserved [SBGC20]. To capture flow features, we use the entropy-based seeding technique of Xu et al. [XLS10]. We normalize the resulting entropy scalar field and treat it as a probability distribution from which to sample positions. In order to not starve low-entropy regions of the flow field,

Table 1: *The dimensions and number of timesteps of each dataset.*

datasets	dimensions ($x \times y \times z$)	timesteps (t)
tornado	$128 \times 128 \times 128$	50
square cylinder	$192 \times 64 \times 48$	100
tangaroa	$300 \times 180 \times 120$	200

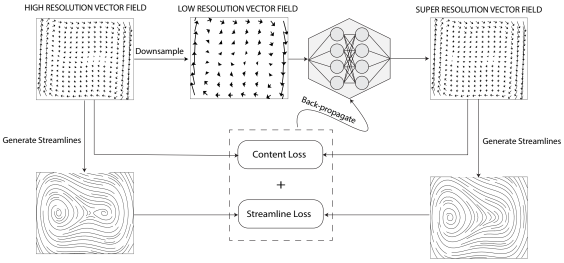


Figure 2: *Overview of our approach. The network takes in the low-resolution vector field as input and outputs a super-resolution vector field. A content loss, alongside a streamline-based loss, between super-resolution and ground truth vector fields are used to optimize the network parameters. The use of 2d vector fields in the figure is for illustrative purpose only.*

we modify the distribution to interpolate between a uniform distribution, one that is a function of the entropy field:

$$s(x) = x\frac{e^x + e^{-x}}{e^{-x} - \alpha e^x}, \tag{4}$$

where $\alpha \in [-1, 0]$ interpolates between distributions.

Integration Length: The length of the streamlines further has an impact on flow features. Specifically, we use Euler Integration with sufficiently small step size and consequently, identify streamline length with the number of steps m taken during integration. The streamline length with which we train can have an impact on the network's ability to generalize, e.g. by training on small-length streamlines, will the network produce vector fields that faithfully reflect long streamlines? Similarly, training on long streamlines may sacrifice the ability to preserve small streamlines.

3.3. Implementation Details

Our network architecture for f closely follows Guo et al. [GYH*20], the only exception that we replace their *voxel shuffle* layers with nearest-neighbor upsampling for simplicity. Through experimentation, we found training the model using Eq. 3 from scratch posed challenges for optimization. Hence, we first train the model using the content loss (Eq. 1) for 25k iterations, and then fine-tune the model using the total loss (Eq. 3) for another 10k iterations. We found Euler Integration with sufficiently small step size and 4^{th} order Runge-Kutta integration give similar results. However, the former trains significantly faster, making it our choice of integration scheme for all the experiments. We use the Adam optimizer [KB14], with a starting learning rate of 10^{-4}, we reduce it by a factor of 0.8 every 1000 iterations until the model fails to improve on withheld validation data. While training, for all the experiments we used a subsampling factor $r = 4$, batch size of 1 and 2,000 streamlines within a batch to form the loss function ($\mathcal{L}_S$).

4. Results

4.1. Dataset and Training Details

All our experiments were carried on the datasets listed in Table 1, where x, y, z represents the spatial dimensions and t represents the number of timesteps in the dataset. To assess generalization of IA-VFS, we include every 4^{th} timestep of a given dataset in the training set and randomly select $t = (\hat{t} \mod 10)$ timesteps for validation, where $\hat{t}$ represents the timesteps not being used for training. All the remaining timesteps are then used for testing purposes. All the experiments were carried out on NVIDIA TESLA V100 GPU.

Baseline We use the following 2 baselines to compare with our technique. (1) **Trilinear Interpolation (TI)**: We use trilinear interpolation to upsample the low-resolution vector field to high-resolution vector field. (2) **Content Loss (CL)**: Using the same network architecture described in Sec. 3.3, we optimize only for the content loss in Eq. 1. Note, this represents Guo et al. [GYH*20], without using an angle-based loss, which we experimentally found to produce similar streamline results.

Evaluation Metric We use two different evaluation metrics to quantitatively evaluate IA-VFS. We use PSNR to evaluate the quality of the super-resolution vector field. PSNR is defined as follows:

$$PSNR(\mathbf{V_h}, f(\mathbf{V_l})) = 20\log_{10} R - 10\log_{10} MSE(\mathbf{V_h}, f(\mathbf{V_l})), \tag{5}$$

R represents the difference between the minimum and maximum value of the vector fields across all the timesteps for a given dataset and $MSE(\mathbf{V_h}, f(\mathbf{V_l}))$ represents the mean square error between the vector fields $\mathbf{V_h}$ and $f(\mathbf{V_l})$.
Since errors accumulate quite easily when calculating streamlines, the position of the last point on a given streamline of $f(\mathbf{V_l})$ can indicate how much it deviated from the last point position of streamline of $\mathbf{V_h}$. To this end, we define ALP (Average last position loss) to evaluate the quality of streamlines as follows:

$$ALP = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{p}_{i,m} - \mathbf{p}'_{i,m}\|_2 \tag{6}$$

where n represents the number of streamlines, $\mathbf{p}_{i,m}$ represents the last point's position of i^{th} streamline of $\mathbf{V_h}$ and $\mathbf{p}'_{i,m}$ represents the last point's position of i^{th} streamline of the $f(\mathbf{V_l})$.

4.2. Quantitative and Qualitative Results

In Table 2, we summarize the quantitative evaluation of IA-VFS against TI and CL by averaging the PSNR (Eq. 5) and ALP (Eq. 6) values across all the test timesteps. We observe that IA-VFS outperforms TI and CL in ALP, indicating that our method more faithfully preserves streamlines. In case of tornado dataset, CL has the highest PSNR but it comes at the cost of lower ALP - this can be seen in Figure 1 where, IA-VFS (b) produces more faithful streamlines as compared to CL (c) which fails to capture the helix like pattern at the eye of the tornado. In Figure 3 we can see the streamline errors made by CL and TI in all the datasets. In square cylinder

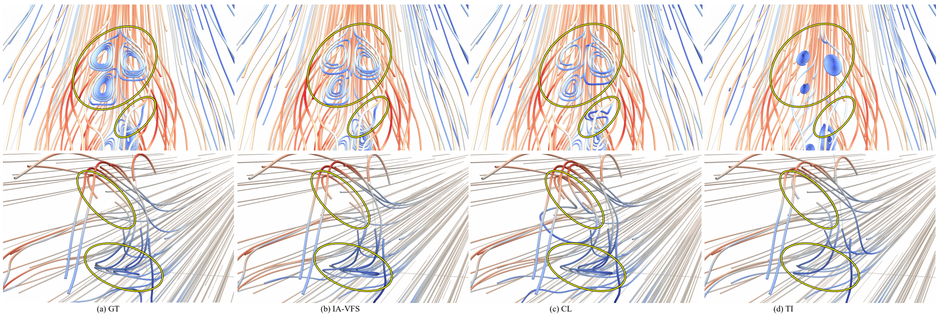


Figure 3: *Comparision of the differences in streamlines of vector fields generated by different models with respect to the ground truth highlighted in yellow. Top to bottom: square cylinder, tangaroa.*

Table 2: *Average last position loss (Eq. 6) and PSNR for all the datasets.*

Dataset	Method	PSNR	ALP
Tornado	TI	47.33	0.233
	CL	**51.69**	0.092
	IA-VFS	50.84	**0.058**
Square cylinder	TI	32.53	0.791
	CL	48.85	0.268
	IA-VFS	**48.86**	**0.245**
Tangaroa	TI	49.68	1.222
	CL	51.96	1.025
	IA-VFS	**52.21**	**0.914**

Table 3: *Average PSNR and ALP values for various α values for tornado dataset.*

α	PSNR	ALP
-1	50.93	0.0372
-0.01	50.88	0.0333
-0.001	50.78	0.0356
-0.0001	50.73	0.0378
0	51.47	0.0513

Table 4: *Average PSNR and ALP values for different λ values for tornado dataset.*

λ	PSNR	ALP
0	51.69	0.0521
0.001	52.04	0.0465
0.1	51.76	0.0379
0.3	51.22	0.0359
0.5	50.88	0.0333
0.7	50.65	0.0356
1	50.33	0.0374

dataset we can see that IA-VFS captures the highlighted streamline whereas CL and TI fails to do so. We can also observe that the spiral flow is more accurate in (b) as compared to (c) and (d).

4.3. Hyperparameter Study

In this section, we analyze how various streamline hyperparameters affect the training process and justify our choices. We experimented with the following hyperparameter settings: the choice of λ (c.f. Eq. 3), streamline seeding, and number of integration steps.

Study of α parameter From Eq 4, we may bias seeds towards high entropy or uniformly-distributed positions via the parameter α. Here we study the influence of α, where $\alpha = -1$ gives us the normalized entropy scalar field back, and increasing α increases the chances of high entropy regions to be selected as seeds.

Table 5: *Average last position loss of streamlines (Eq. 6) for models trained and evaluated on different streamline lengths.*

Train \ Eval	150	200	250	300	350	400
150	0.0262	0.0357	0.0454	0.0552	0.0657	0.0764
250	0.0260	0.0347	0.0438	0.0531	0.0630	0.0732
300	0.0262	0.0343	0.0427	0.0514	0.0605	0.0699
350	0.0261	0.0337	0.0419	0.0501	0.0588	0.0678
400	0.0264	0.0342	0.0424	0.0508	0.0596	0.0686

In Table 3, we observe that there is a trade-off between PSNR and ALP for the tornado dataset based on the value of α. Heavily biasing towards the high entropy regions ($\alpha = [-0.0001, 0]$) leads to high ALP values. Since the network receives few important streamlines it fails to capture them accurately during evaluation. Meanwhile, a more spread out selection of seed points in and around high entropy regions with $\alpha = -0.01$ gives us the best ALP value and with acceptable PSNR.

Study of λ hyperparameter We can see in Equation 3 that λ controls the weight on the content loss and streamline loss. From Table 4 we observe that as we increase the value of λ we see a decrease in both PSNR and ALP. We found that a $\lambda = 0.3$ provides a good balance between PSNR and ALP.

Study of streamline length Streamline length determines the maximum number of steps to be taken while integrating the streamline. From Table 5 we can see that models trained on longer streamlines e.g. 300, 350 and 400 outperform models trained on smaller streamlines e.g. 150, 250 in terms of average ALP values. We found that model trained on streamline length of 350 provides good generalization when evaluated across different streamline lengths.

5. Conclusion and Future Work

In this work, we proposed an integration-aware super resolution technique for 3d vector fields. We think our approach is an important steps towards incorporating visualization aspects of vector fields in the optimization process. We show the benefits of using our technique and how various factors of vector visualization via streamlines affects the training process. There are several directions we would like to explore for our future work. In this work we experimented with a downsampling factor of 4, and we intend to try our framework on larger scaling factor. We have thus far, only considered spatial super-resolution of vector fields, and we intend to take into account the temporal coherence of unsteady vector fields.

References

[BLL18] Berger, Matthew, Li, Jixian, and Levine, Joshua A. "A generative model for volume rendering". *IEEE transactions on visualization and computer graphics* 25.4 (2018), 1636–1650.

[CCJ*18] Cheng, Hsueh-Chien, Cardone, Antonio, Jain, Somay, et al. "Deep-learning-assisted volume visualization". *IEEE transactions on visualization and computer graphics* 25.2 (2018), 1378–1391.

[DHX*09] Dai, Shengyang, Han, Mei, Xu, Wei, et al. "Softcuts: a soft edge smoothness prior for color image super-resolution". *IEEE transactions on image processing* 18.5 (2009), 969–981.

[DLHT15] Dong, Chao, Loy, Chen Change, He, Kaiming, and Tang, Xiaoou. "Image super-resolution using deep convolutional networks". *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), 295–307.

[DLT16] Dong, Chao, Loy, Chen Change, and Tang, Xiaoou. "Accelerating the super-resolution convolutional neural network". *European conference on computer vision*. Springer. 2016, 391–407.

[Duc79] Duchon, Claude E. "Lanczos filtering in one and two dimensions". *Journal of Applied Meteorology and Climatology* 18.8 (1979), 1016–1022.

[GYH*20] Guo, Li, Ye, Shaojie, Han, Jun, et al. "SSR-VFD: Spatial super-resolution for vector field data analysis and visualization". *2020 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE Computer Society. 2020, 71–80.

[HCL*18] Han, Wei, Chang, Shiyu, Liu, Ding, et al. "Image super-resolution via dual-state recurrent networks". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 1654–1663.

[HTW18] Han, Jun, Tao, Jun, and Wang, Chaoli. "FlowNet: A deep learning framework for clustering and selection of streamlines and stream surfaces". *IEEE transactions on visualization and computer graphics* (2018).

[HTZ*19] Han, Jun, Tao, Jun, Zheng, Hao, et al. "Flow Field Reduction Via Reconstructing Vector Data From 3-D Streamlines Using Deep Learning". *IEEE computer graphics and applications* 39.4 (2019), 54–67.

[HW19] Han, Jun and Wang, Chaoli. "TSR-TVD: Temporal super-resolution for time-varying data analysis and visualization". *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2019), 205–215.

[HWG*19] He, Wenbin, Wang, Junpeng, Guo, Hanqi, et al. "InSituNet: Deep image synthesis for parameter space exploration of ensemble simulations". *IEEE transactions on visualization and computer graphics* 26.1 (2019), 23–33.

[JAF16] Johnson, Justin, Alahi, Alexandre, and Fei-Fei, Li. "Perceptual losses for real-time style transfer and super-resolution". *European conference on computer vision*. Springer. 2016, 694–711.

[KB14] Kingma, Diederik P and Ba, Jimmy. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980* (2014).

[Key81] Keys, Robert. "Cubic convolution interpolation for digital image processing". *IEEE transactions on acoustics, speech, and signal processing* 29.6 (1981), 1153–1160.

[KKM16] Kim, Jiwon, Kwon Lee, Jung, and Mu Lee, Kyoung. "Deeply-recursive convolutional network for image super-resolution". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 1637–1645.

[LHAY17] Lai, Wei-Sheng, Huang, Jia-Bin, Ahuja, Narendra, and Yang, Ming-Hsuan. "Deep laplacian pyramid networks for fast and accurate super-resolution". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 624–632.

[LSK*17] Lim, Bee, Son, Sanghyun, Kim, Heewon, et al. "Enhanced deep residual networks for single image super-resolution". *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, 136–144.

[LTH*17] Ledig, Christian, Theis, Lucas, Huszár, Ferenc, et al. "Photo-realistic single image super-resolution using a generative adversarial network". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 4681–4690.

[MO08] Marquina, Antonio and Osher, Stanley J. "Image super-resolution by TV-regularization and Bregman iteration". *Journal of Scientific Computing* 37.3 (2008), 367–382.

[SBGC20] Sane, Sudhanshu, Bujack, Roxana, Garth, Christoph, and Childs, Hank. "A Survey of Seed Placement and Streamline Selection Techniques". *STAR* 39.3 (2020).

[SCH*16] Shi, Wenzhe, Caballero, Jose, Huszár, Ferenc, et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 1874–1883.

[SCI18] Shocher, Assaf, Cohen, Nadav, and Irani, Michal. ""zero-shot" super-resolution using deep internal learning". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 3118–3126.

[SSH17] Sajjadi, Mehdi SM, Scholkopf, Bernhard, and Hirsch, Michael. "Enhancenet: Single image super-resolution through automated texture synthesis". *Proceedings of the IEEE International Conference on Computer Vision*. 2017, 4491–4500.

[ST19] Shi, Neng and Tao, Yubo. "CNNs based viewpoint estimation for volume visualization". *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.3 (2019), 1–22.

[SXS08] Sun, Jian, Xu, Zongben, and Shum, Heung-Yeung. "Image super-resolution using gradient profile prior". *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, 1–8.

[TLLG17] Tong, Tong, Li, Gen, Liu, Xiejie, and Gao, Qinquan. "Image super-resolution using dense skip connections". *Proceedings of the IEEE International Conference on Computer Vision*. 2017, 4799–4807.

[TYL17] Tai, Ying, Yang, Jian, and Liu, Xiaoming. "Image super-resolution via deep recursive residual network". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 3147–3155.

[XFCT18] Xie, You, Franz, Erik, Chu, Mengyu, and Thuerey, Nils. "tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow". *ACM Transactions on Graphics (TOG)* 37.4 (2018), 1–15.

[XLS10] Xu, Lijie, Lee, Teng-Yok, and Shen, Han-Wei. "An information-theoretic framework for flow visualization". *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), 1216–1224.

[YXYN15] Yan, Qing, Xu, Yi, Yang, Xiaokang, and Nguyen, Truong Q. "Single image superresolution based on gradient profile sharpness". *IEEE Transactions on Image Processing* 24.10 (2015), 3187–3202.

[ZHW*17] Zhou, Zhenglei, Hou, Yule, Wang, Qirui, et al. "Volume upscaling with convolutional neural networks". *Proceedings of the Computer Graphics International Conference*. 2017, 1–6.

[ZTK*18] Zhang, Yulun, Tian, Yapeng, Kong, Yu, et al. "Residual dense network for image super-resolution". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 2472–2481.