The U-Net based GLOW for
Optical-Flow-Free Video Interframe Generation

Saem Park[1,2][a], Donghoon Han[1][b] and Nojun Kwak[1][c]

[1]*Seoul National University, Seoul, South Korea*
[2]*LG Electronics, Seoul, South Korea*

Abstract: Video frame interpolation is the task of creating an interframe between two adjacent frames along the time axis. So, instead of simply averaging two adjacent frames to create an intermediate image, this operation should maintain semantic continuity with the adjacent frames. Most conventional methods use optical flow, and various tools such as occlusion handling and object smoothing are indispensable. Since the use of these various tools leads to complex problems, we tried to tackle the video interframe generation problem without using problematic optical flow. To enable this, we have tried to use a deep neural network with an invertible structure, and developed an U-Net based Generative Flow which is a modified normalizing flow. In addition, we propose a learning method with a new consistency loss in the latent space to maintain semantic temporal consistency between frames. The resolution of the generated image is guaranteed to be identical to that of the original images by using an invertible network. Furthermore, as it is not a random image like the ones by generative models, our network guarantees stable outputs without flicker. Through experiments, we confirmed the feasibility of the proposed algorithm and would like to suggest the U-Net based Generative Flow as a new possibility for baseline in video frame interpolation. This paper is meaningful in that it is the new attempt to use invertible networks instead of optical flows for video interpolation.

1 INTRODUCTION

In the introduction, we want to explain the necessity of generating video interframes, existing methods of using deep neural networks, general optical flow problems, and the basics of invertible networks.

1.1 The Necessity of Generating Video Interframes

Interframe generation, which creates an intermediate frame using the information of two consecutive frames, is one of the techniques frequently used in TV systems. Most input video sources have a refresh rate of 24, 30, and 60 Hz per second, but TV can often output frames at a refresh rate of 120 Hz per second. To achieve this, the TV interpolates over the temporal phase to form an intermediate image, producing a high frame-rate video of 120 Hz. Because the simple synthesis of the front and rear frames cannot produce smooth video and the accompanying judder is inevitable, the interframe generation algorithms generally move the front or rear frames in the motion vector direction using optical flow. Optical flow (Horn and Schunck, 1981; Barron et al., 1994) is the information indicating in which direction each pixel is moving in a frame by using the correlation between the front and rear frames. Traditionally, a man-made program was used to calculate the optical flow and algorithms use it to create an intermediate image.

1.2 Video Interpolation Method using DNN

In the DNN field, several attempts have been made to generate interframes since around 2017. One attempt was interpolating the front and rear frames by training a network to generate a supervised optical flow like Flow-net (Dosovitskiy et al., 2015a; Ilg et al., 2017a) and PWC-Net (Sun et al., 2018). The other attempt was Super SloMo (Jiang et al., 2018) which can create intermediate frames by learning the unsupervised

---

[a] https://orcid.org/0000-0002-9727-4272
[b] https://orcid.org/0000-0002-3781-0256
[c] https://orcid.org/0000-0002-1792-0327

(a) Limitation of optical flow (b) Limitation of pixel shuffle

Figure 1: **Limitations of previous works.** (a) shows a part of FlyingThings3D (Mayer et al., 2016) generated by warping the previous frame using the ground truth optical flow. The object appears to be broken due to occlusion issues despite the ground truth being applied. (b) shows the interim results of learning with GLOW (Kingma and Dhariwal, 2018) instead of UGLOW. Pixel shuffle and 1x1 convolution cause side effects like the Bayer pattern.

optical flow using high-speed framed video shot with a high-speed camera.

In the former case, since learning is possible only with supervised information on optical flow, there is an inconvenience of making motion information for each video in a frame unit, which causes difficulties in generating training data. This is a costly job and the models have to struggle with limited training data.

In the latter case, only the front and rear frames are used as inputs, and the optical flow is predicted by itself without additional supervised optical flow information. This method does not require any additional information about the optical flow in the training data, so the training can be performed with any video input. In the paper of Super SloMo, (Jiang et al., 2018) the network was trained mainly using slow-motion videos with a frame rate above 240Hz, which has several advantages. First of all, high frame rated videos carry much more frames than normal videos in a fixed time. It means that videos move smoother and carry more contextual information. Taking the advantage of high frame-rated videos, the model can boost performance in predicting the context between the frames. Also, the video is relatively clearly divided into foreground and background, and the majority of movements are centered on horizontal and vertical moves rather than complex motions.

1.3 Limitations of using Optical Flow

As described above, most interframe algorithms use optical flow to move the front and rear frames to create intermediate frames. However, the use of optical flow cannot produce perfect results even when the ground truth is applied due to various problems.

There are also many methods to use deep learning on video interframe generation (Niklaus et al., 2017; Meyer et al., 2018; Peleg et al., 2019) but they cannot be free from optical flow problems. The problems of optical flow are as follows. The first is that it cannot accurately handle the boundaries of each object, and the second is that it is difficult to handle the occluded area.

The former boundary problem arises from the fact that when estimating the motion vector of an object, the boundary between the object and the background cannot be accurately known. In particular, if the boundary is predicted to be smaller than the object, it will be cut, causing serious problems. For this reason, the motion vector of an object is generally set to erode the background more widely. However, if the background is eroded too much, it can cause side effects such as halo. Also, when calculating the optical flow of the entire image, in some cases, not all pixels are filled and there may be empty holes. The question arises as to how to interpolate these holes. In this case, when a simple averaging is used, a motion vector that does not exist in the actual video is generated, and when the surrounding value is expanded, the boundary becomes more visible. Another complex case is when there are multiple motion vectors within an object. A common example is when an object rotates. In this kind of circumstance, the boundary of the motion is unclear and the object appears to be broken.

The latter occlusion is the processing problem of overlapping areas. This is a more difficult problem than the former one. It is because there is no correlation between the foreground and the background when the generator tracks movements. The first process of treating occluded areas is to find out which one is ahead. Objects coming forward are taken from the next frame and the background should be deleted. However, it is not easy to predict what will come forehead when objects overlap each other. Even if the prediction is made, it is very difficult to create an intermediate image with only limited (temporal) directional data. The occlusion handling that covers the optical flow problem is very difficult but important, and MaskFlownet (Zhao et al., 2020) which covers the problems of Flownet (Dosovitskiy et al., 2015b; Ilg et al., 2017b) drew many attentions last year.

Apart from several difficulties mentioned above, there are even more issues such as not being able to find motion in repetitive patterns or missing small objects. These drawbacks make it impossible for optical flow to generate a clear answer. Looking at the image generated by applying the ground truth of the optical flow to the previous frame in Fig. 1(a), it is easy to understand why we do not want to use it.

In this paper, we propose the U-Net based Generative Flow that generates clean optical results. Without using optical flow, U-Net based Generative Flow is expressive enough to resemble the operations of optical flow if necessary and even outperforms the previous approaches.

1.4 Invertible Network

To create an intermediate image without using an optical flow, we considered the use of an invertible network. In general DNN, due to the non-linear function and pooling process, it is impossible to perform reverse processing from the latent space to the original input again. In this case, there is no way to recover the original input from latent space due to data loss.

There are algorithms such as GLOW (Kingma and Dhariwal, 2018), MintNet (Song et al., 2019), BayesFlow (Radev et al., 2020), and i-RevNet (Jacobsen et al., 2018a; Jacobsen et al., 2018b) developed to prevent data loss in latent space in DNN. This bypasses half of the input channel and sends it to the next layer. This is used to generate the same convolution result as the forward path in the inverse path and restore the original value using this. The idea of GLOW is that interpolation over a latent space can produce an intermediate image between two images that have semantic continuity. Our idea is that interpolating between two individual video frames similarly produces an intermediate frame with semantic continuity.

The invertible network has several advantages. The first point is that the image quality is not compromised and the original restoration is guaranteed so that the original quality can be maintained even for videos with sufficiently high input resolution. The second advantage is that unlike generative models such as GAN(Goodfellow et al., 2014), a 1:1 functional relationship between an image and a point in a latent space is established. This ensures the one and only intermediate image for each latent space without random occurrence, enabling video processing without flickering.

1.5 Limitations of Existing Invertible Network

We want to interpolate between frames by applying an invertible network to a video. For this, some of the problems of invertible networks had to be solved. The biggest problem among them is that invertible networks such as FLOW and i-RevNet increase the number of channels while reducing the image size through pixel shuffle. This creates a fatal problem of

generating Bayer patterns during interpolation, which can be seen in Fig. 1(b). We wanted the inter-pixel convolution process to look smooth enough for image processing, and for this, we had to remove the pixel shuffle. However, simply removing the pixel shuffle does not have a way to increase the number of channels while maintaining the amount of information contained in the input RGB channels, so the final network output also ends up with three channels. In this case, generated information is also limited to three channels unless some measures like downsampling are considered. Another problem is that FLOW uses 1x1 convolution to propagate information in half of the channels, but it is difficult to divide the three channels of RGB in half. Also, since the 1x1 convolution does not refer to the surrounding pixels, only the data of the channel combined with the pixel shuffle is calculated. In image processing, local convolution is desirable because it determines the type and shape of an object by referring to surrounding pixels, and the 1x1 convolution is not suitable for this kind of use. I-RevNet also uses pixel shuffle and its effect seemed vain. To sort out this issue, we have devised an U-Net based Generative Flow suitable for such image processing.

2 CONTRIBUTION

We introduce a DNN model suitable for generating video interframes using an invertible convolution network. The proposed network expands the number of channels without pixel shuffle and performs local convolution processing. Also, using U-Net, it is possible to refer to the upper layer's information in the form of a pyramid and take the advantage of being invertible. Also, we propose a new learning method for video interpolation between frames. Since the network can learn the input image and generate the necessary information itself, the network can be trained in an unsupervised manner. With the generated information, no data other than latent space is needed to generate the intermediate image through linear interpolation.

The contributions of this paper are as follows:

1. The proposed method is the world's first attempt to suggest a new approach for video frame interpolation using an invertible deep neural network. Since no optical flow is used in our method, we can fundamentally avoid the problems that stem from optical flow.

2. We propose a novel U-Net based Generative Flow (UGLOW) that is invertible while utilizing lo-

cal convolution and U-Net structure without pixel shuffle.

3. Using the proposed UGLOW, we also propose a training method that transforms continuous video frames into a linear latent space on the time axis.

3 METHOD

3.1 Concept

In this section, we propose the concept of U-Net based Generative Flow and its two major methodologies. The idea of our concept is as follows. When generating an intermediate frame, the easiest way would be the linear interpolation using two adjacent frames as inputs. However, such a method only produces overlay images that do not help with judder. To generate a sophisticated intermediate frame, linear interpolation should be performed on a transformed space after projecting adjacent frames onto it. Then, a linear relationship with transformed space is established.

However, there are several conditions for such linear interpolation to perform as expected. First, it should be possible to restore the original image from the converted space. Second, the transformed space should have a linear relationship with the time axis. To meet those conditions, the model should guarantee invertible non-linear conversion between the frame and the converted space.

The first condition got its idea from GLOW (Kingma and Dhariwal, 2018). As you can see in Fig. 2, GLOW smooths out interpolations between the two faces. We can think of it as an interpolation between two video frames. An invertible network like GLOW was an ideal candidate for intermediate frame generation because the latent space has a non-linear relationship with the input image. On top of that, the original image can be restored from the latent space. The second condition can be met if a model can generate the second frame's latent space with the interpolation of latent spaces from the first and third frame.

We suggest the two novel methods to meet the concept and conditions. The first is the U-Net based Generative Flow (UGLOW), which is specially designed to learn video information in the most advantageous manner. In the latter part, we introduce the very original sub-modules that make up the network, how it is made to be invertible, and how the whole network is structured. Details of UGLOW are described in section 3.2. The second is the loss metric that enables the UGLOW to generate plausible frames without complex algorithms. Here we would like to intro-

duce the idea used to make the latent space linear on the time axis. The detailed learning method will be introduced in sections 3.3 and 3.4.

3.2 U-Net based Generative Flow (UGLOW)

We needed to devise an invertible network that can grasp the context between the frames to replace optical flow. In some cases, a generator may have to track a large number of pixels depending on the size and movement of an object in the frame. If this is the case, it would be expensive to cover the whole area with convolution with a large kernel size. To address this issue, we tried to use an efficient network and U-Net was a feasible option. On the contracting path of U-Net, as the network propagates forward, subsampling enables the network to track a wider area with the same sized kernel. To be exact, each layer's subsampling enables kernel to refer twice a larger area. At the same time, skip connection is used to transmit information from the contracting path to the expansive path without loss, which prevents information loss due to the size of the bottleneck in the middle of the U-Net. Even in optical flow algorithms, pyramid structures with each layer halved in size are often used to detect large objects or fast motion vectors. Therefore, the U-Net can be said to be a suitable structure for an intermediate frame generation task. In addition, we developed the idea of skip connection and made the network invertible.

As mentioned in the introduction, the invertible network uses pixel shuffle and 1x1 convolution which can aggravate the output. U-Net based Generative Flow uses 3x3 or 5x5 convolution to refer to local information and works without pixel shuffle. Every U-Net consists of 4 down-blocks, 1 mid-block, and 4 up-blocks, each consisting of 2 convolution layers and Leaky-ReLU. The last output of the U-Net uses sigmoid as an activation so that it could adjust the image input normalized to 0-1. The number of channels is preserved by removing the pixel shuffle, so we need to expand the output channel to increase the information. For this reason, it was necessary to make two types of blocks in which the channels are extended or maintained according to the number of input and output channels. We propose two sub-blocks that modified U-Net to an invertible form.

The first is a channel expend block in which the output has a channel twice as large as the input. This was designed to increase the number of channels as it is not enough to generate various information with only 3-channel input. (a-1) and (a-2) in Fig.3 correspond to this. Since UGLOW is an invertible struc-
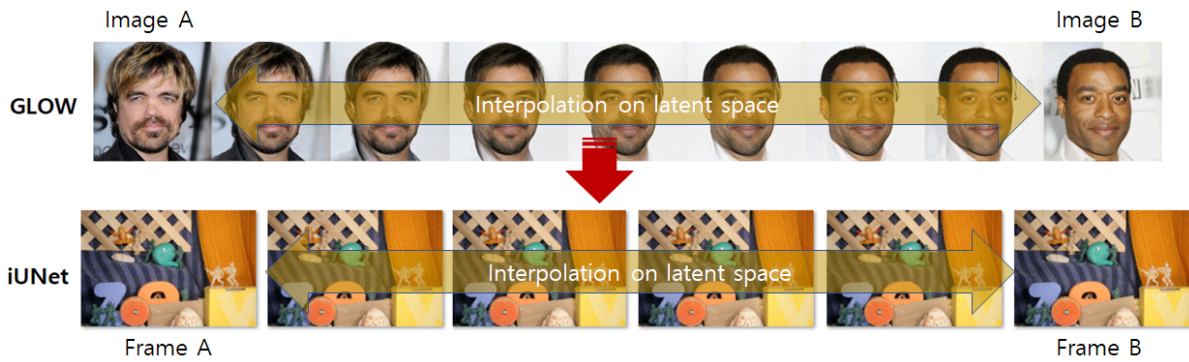
Figure 2: **The concept of frame interpolation using UGLOW** The proposed UGLOW is a concept that interpolates video frames in latent space, like invertible networks such as GLOW.

(a-1) Channel Extend Module (Forward)

(a-2) Channel Extend Module (Backward)

(b-1) Channel Maintain Module (Forward)

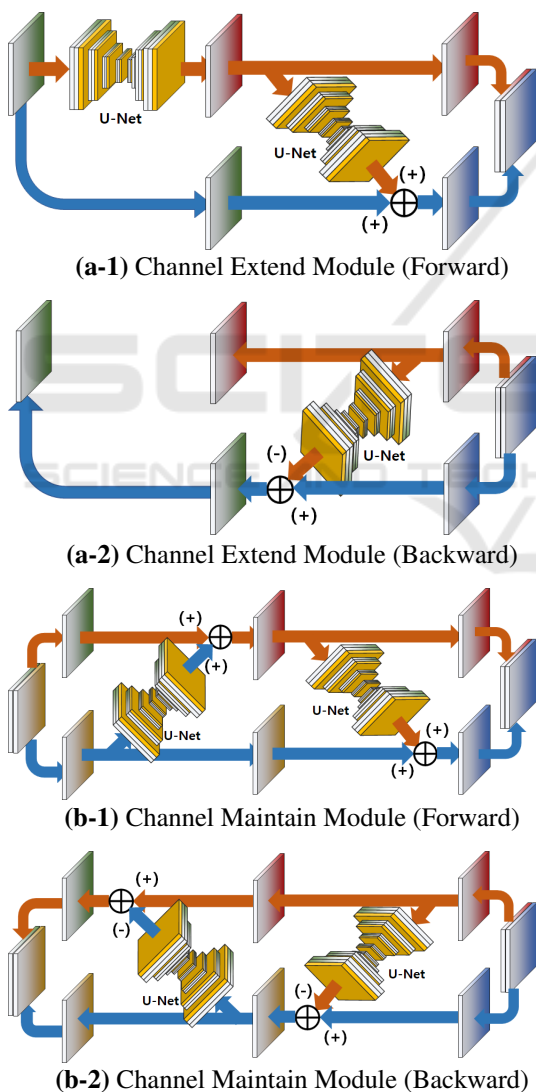**(b-2)** Channel Maintain Module (Backward)

Figure 3: **U-Net based Generative Flow Module.** These are invertible module using U-Net. Two types are depending on whether the channel is amplified or not, each showing how to recover the input from the output.

ture, of course, when reversed, the channel is reduced by half. Looking at (a-1) in Fig.3, it can be divided into the first half and the second half. The first half is to double the channel by attaching the original to the output of the U-Net like a skip connection, and it can be reversed by using the skip connected data as it is. In the second half, the data from the other side is added via U-Net, so all the inputs are adjusted to a non-linear transformation. The output created in this way can be restored by (a-2). Since the output of the U-Net in the second half has the same value in the forward and reverse path, the input can be restored by simple subtraction instead of addition.

The second is a channel maintain block with the same number of channels as outputs and inputs, as shown in Fig.3 (b-1) and (b-2). This block was designed to retain the number of channels. By using both extend block and maintain block, layers can be stacked deep enough while having enough channels. Similar to the existing FLOW, this module uses half of the channel to process the other half and performs it in the opposite direction again, making it invertible while maintaining the number of channels. Each U-Net has a quad down-block, a middle-block, and a quad up-block inside. As shown in Fig.4, we designed a deep neural network that has more than 100 convolution layers by stacking 11 invertible modules. We name it UGLOW, meaning U-Net based Generative Flow.

3.3 Definition of Loss for Learning

First, prepare a total of 3 consecutive frames from any video. These frames become training data, and even with a small number of videos, you can get a large number of training data with frame sliding. Network learning proceeds with the simplest ideas. The core of the proposed learning method is to make sure that the three consecutive frames, whose center frame is
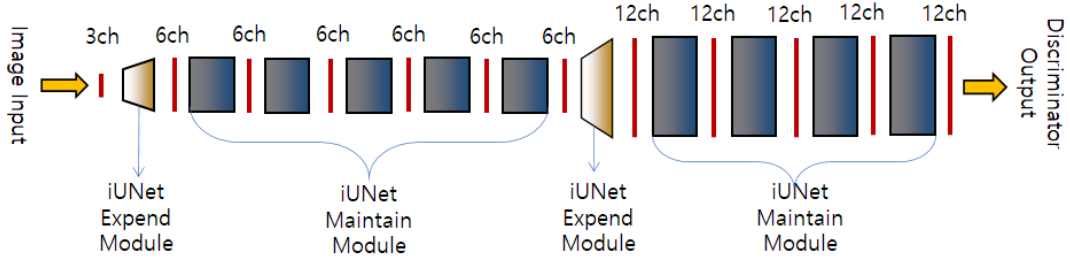
Figure 4: **U-Net based Generative Flow architecture.** This is the overall structure of the UGLOW used in this paper.
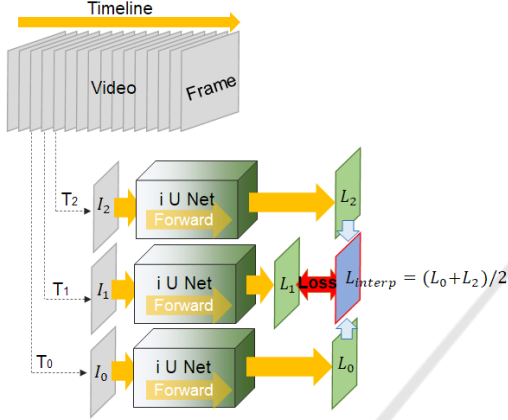

Figure 5: **Concept of the loss in the latent space.** It makes latent spaces from consecutive frames have a linear relationship.
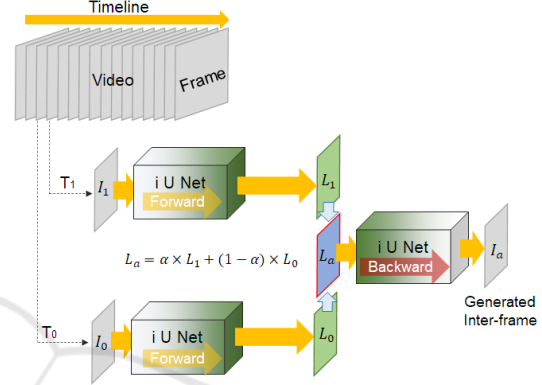

Figure 6: **Concept of the loss in the input space.** It makes reverse generated interframe as original like.

generated from the other two through UGLOW, have a linear relationship. We suggest that simple linear blending in the latent space created through this learning method can represent the intermediate frame well in the time axis between the two frames.

Specifically, the input data of three consecutive frames are named I_0, I_1, I_2 and each image is transformed into a latent space via a reversible network.

Figure 7: **How to make interframe.** This figure shows how interframe generation works. When a linear relationship between the latent space and the time axis is established, it is possible to create a frame at an arbitrary time position with only alpha blending.
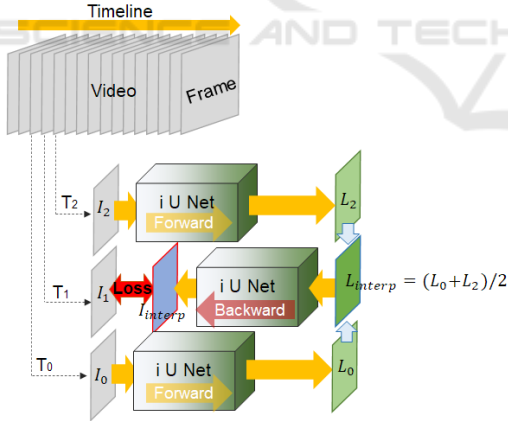
The latent spaces created in this way are named L_0, L_1, L_2. At this time, the loss is defined to minimize the difference between L_1 and L_{inter} which is created by linearly interpolating L_0 and L_2. The UGLOW we used can restore I from L through an inverse processing, and there is non-linearity between I and L. This enables our network to minimize $||\text{model.reverse}(L_{inter}) - I_1||^2$ and $||L_{inter} - L_1||^2$ at the same time.

In our algorithm, the loss metric is designed to optimize two tasks.

1. The loss in the latent space aims to minimize the difference between the result of interpolation on the latent space and the latent space created from the intermediate frame: $Loss_L$. When enough learning is done, the latent spaces show a linear relationship with each other on the time axis. (Fig.5)

2. The loss in the input space aims to minimize the difference between the image I_{inter} restored from L_{inter} and I_1: $Loss_I$. When enough learning is done, the reversed intermediate frame will match the actual intermediate frame. (Fig.6)

The detailed formula for the discriminator loss can

be defined as follows.

$$\text{model} = \text{UGLOW}(I) \qquad (1)$$

$$\begin{aligned} L_0 &= \text{model}(I_0) \\ L_1 &= \text{model}(I_1) \qquad\qquad (2) \\ L_2 &= \text{model}(I_2) \end{aligned}$$

$$L_{inter} = (L_0 + L_2)/2 \qquad (3)$$

$$Loss_L = ||L_{inter} - L_1||^2. \qquad (4)$$

The detailed formula for the reconstruction loss can be defined as follows.

$$I_{inter} = \text{model.reverse}(L_{inter}) \qquad (5)$$

$$Loss_I = ||I_{inter} - I_1||^2. \qquad (6)$$

The final loss can be defined as follows.

$$Loss = w_L \times Loss_L + w_I \times Loss_I, \qquad (7)$$

where w_L and w_I are values for weight adjustment for each loss. When learning UGLOW by combining these two losses, UGLOW learns how to set up a continuous frame to be a linear relationship in the latent space. Due to the linear relationship of latent spaces on the time axis, simple blending can produce the result of an arbitrary mid-point without optical flow (Fig.7). Also, by ensuring inverse restoration, the image produced by the interpolated discriminator guarantees the same quality as the actual intermediate frame.

3.4 Training Method and Settings

The proposed training method has two main types. The first is the offline training method, which trains the network using the entire training set, and then generates interframes with these pre-trained parameters. The second method is the online training method that fine-tunes network using nearby frames with target frames. In this case, since additional training with the inputs must be performed to create each output, the training cost is much higher. However, as the online training method enables the network to refer to nearby frames more thoroughly, it allows the network to better handle difficult tasks like accelerated motions, occluded areas, and complex motions. In the former case, it can be used when there is a limitation on the cost. In the latter case, it is used when a better result is needed without a cost limit.

The training was performed with the Middlebury dataset(Baker et al., 2011), which is commonly used to see the performance of optical flow. The Middlebury dataset consists of 11 videos for training and provides the same number of videos for evaluation. Each video consists of 8 consecutive frames, and the optical flow ground truth is provided only in the training set. However, we did not refer to this ground truth at all which differentiates our network from other conventional approaches.

Offline training was conducted only with the Middlebury training set, and the network was trained to reduce $Loss_L$ and $Loss_I$ using consecutive 3 frames made by the frame sliding method. Because the frame-sliding method is adopted, 6 sets of inputs per video, and 66 training data were used as the training set in total. We trained the entire training set with 200 epochs and saw this as the result of offline learning. As a hyperparameter for training, the initial LR was 0.1, LR decays by 0.95 times for each epoch, and SGD was used as an optimizer. The weights we used for w_L and w_I are 0.1 and 1.0. The w_I is greater because restoring a frame is our primary purpose, not the discriminator.

Online training starts with pre-trained offline parameters and additional training is performed for each evaluation video. Specifically, since the Middlebury evaluation set checks the inferred frame of the 10th frame, we trained the network with 1500 iterations using two training samples: frame 7-8-9 and 11-12-13. For a fair evaluation of our novel method, the 10th frame was not included in the training.

4 RESULT

In this section, we are going to evaluate the performance of the proposed method by generating the 10th frame of the Middlebury evaluation set. There are two kinds of evaluation. The former is an empirical evaluation that visually compares actual results, the latter is objective evaluation such as PSNR and SSIM that measures the difference from the original.

4.1 Empirical Evaluation

Fig 8 shows some examples of our experiment. The image on the left is ground truth, and the image in the middle shows the middle frame transformed by linear blending in latent space by entering frames 9 and 11 of the Middlebury evaluation set. The image on the right is the result of performing the same linear interpolation in image space, not in latent space. In the image on the right, we can see how fast the object is moving between the two frames. In latent space, the proposed method combines two distant objects and complex details. Although optical flow output may seem more elaborate, our U-Net based Generative Flow has its originality in the methodology.
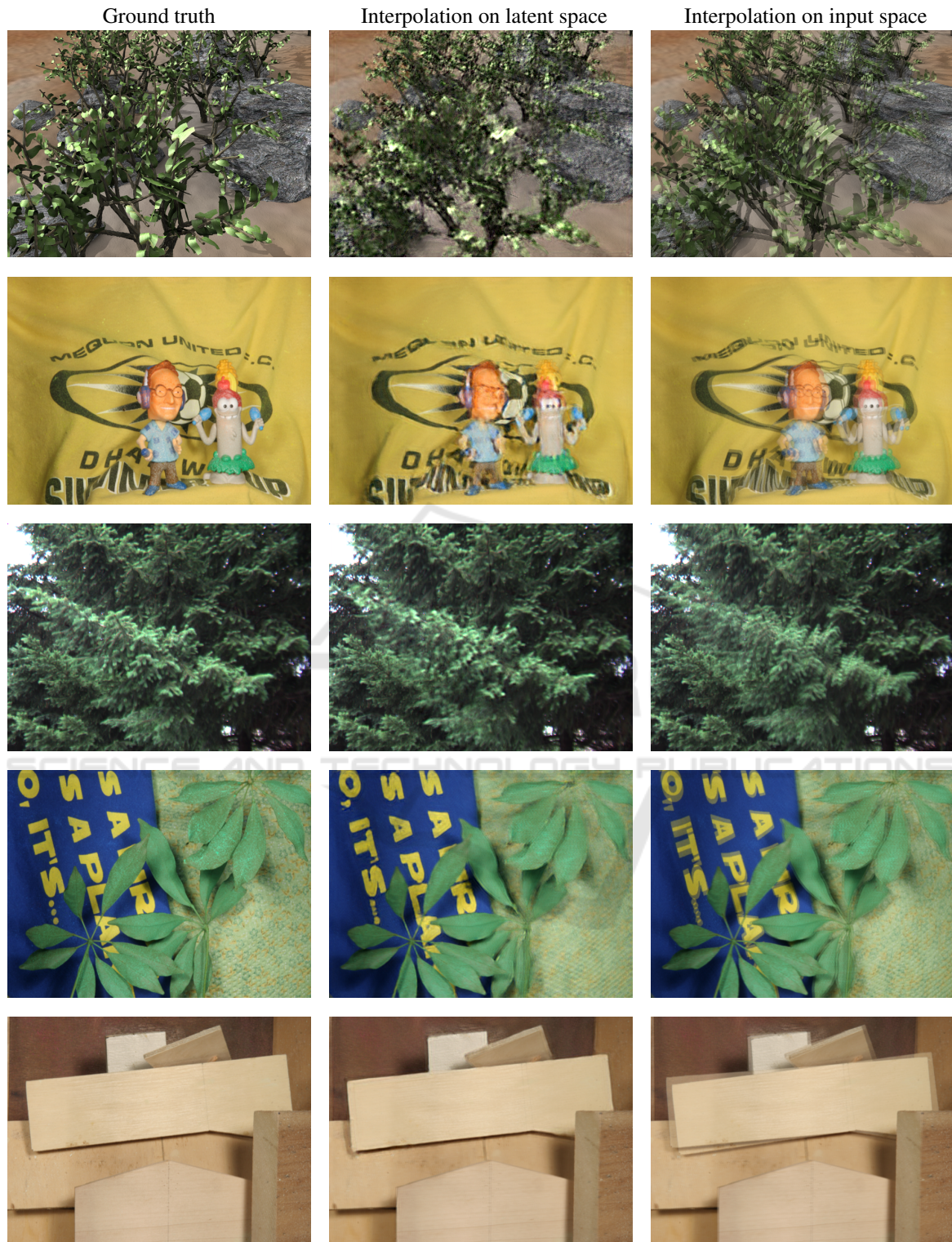
Figure 8: **Interpolation result on latent sapce vs input space** The middle row is an intermediate frame created with only linear interpolation in the learned latency space of UGLOW without using any optical flow at all. Unlike blending in the input space on the right, the proposed method makes a frame corresponding to the actual middle position.

Table 1: **PSNR and SSIM.** This table shows the experimental results of PSNR and SSIM, which reproduced frame 10 of the Middlebury evaluation set using only frames 9 and 11. The proposed algorithm shows improved evaluation values compared to simple image blending in all videos.

Video Name	PSNR (ref)	PSNR (ours)	SSIM (ref)	SSIM (ours)
Grove	15.90	16.77	0.249	0.322
Mequon	23.22	25.06	0.738	0.797
Yosemite	27.11	29.40	0.774	0.845
Dumptruck	24.70	25.02	0.916	0.920
Wooden	27.16	32.43	0.852	0.890
Army	33.89	34.92	0.930	0.932
Basketball	23.98	25.96	0.852	0.876
Evergreen	23.35	24.52	0.778	0.811
Backyard	22.08	23.26	0.688	0.705
Schefflera	25.55	26.81	0.654	0.696
Urban	23.00	25.13	0.600	0.641
Average	24.54	26.30	0.730	0.767

The U-Net based Generative Flow only uses linear interpolation, unlike other conventional approaches that rely on optical flow. Another great advantage is that our outputs showed similarities with the outputs made with simple blending on difficult tasks such as the face of the doll, leaves from the background, and the texture of the stones. This helps guarantee stable frames when processing video.

4.2 Objective Evaluation

For an objective comparison, we measured the difference between our result and the 10th frame from the Middlebury evaluation set. PSNR and SSIM were used here. The results show higher values, as shown in the Table1.

5 CONCLUSION

In this paper, we proposed a new method of generating intermediate frames using video data itself without making optical flow information using an invertible deep neural network. We proposed UGLOW, a reversible network that produces better results, and confirmed its feasibility using the Middlebury data set. We developed a loss that induces a temporal linear relationship between successive frames of video in a latent space and proposed an algorithm capable of generating mid-view results using a trained reversible network. We have shown that this intuitive approach made plausible results through empirical and objective measures.

The biggest contribution of our proposal is that it

is the first attempt not to use optical flow for video interpolation. This aligns with the paradigm that deep learning can learn everything without relying on a knowledge-based system. As future works, we will verify this proposal in various test sets and improve the performance to be similar to the model using optical flow.

ACKNOWLEDGEMENTS

REFERENCES

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31.

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015a). Flownet: Learning optical flow with convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., and Brox, T. (2015b). Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*, 17(1-3):185–203.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017a). Flownet 2.0: Evolution of optical flow estimation with deep networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017b). Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470.

Jacobsen, J., Smeulders, A. W. M., and Oyallon, E. (2018a). i-revnet: Deep invertible networks. *CoRR*, abs/1802.07088.

Jacobsen, J.-H., Smeulders, A., and Oyallon, E. (2018b).

i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*.

Jiang, H., Sun, D., Jampani, V., Yang, M.-H., Learned-Miller, E., and Kautz, J. (2018). Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 10215–10224. Curran Associates, Inc.

Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2016). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. arXiv:1512.02134.

Meyer, S., Djelouah, A., McWilliams, B., Sorkine-Hornung, A., Gross, M., and Schroers, C. (2018). Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 498–507.

Niklaus, S., Mai, L., and Liu, F. (2017). Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679.

Peleg, T., Szekely, P., Sabo, D., and Sendik, O. (2019). Im-net for high resolution video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2398–2407.

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., and Köthe, U. (2020). Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Song, Y., Meng, C., and Ermon, S. (2019). Mintnet: Building invertible neural networks with masked convolutions. *arXiv preprint arXiv:1907.07945*.

Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. (2018). Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943.

Zhao, S., Sheng, Y., Dong, Y., Chang, E. I.-C., and Xu, Y. (2020). Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.