A Flexible Recurrent Residual Pyramid Network for Video Frame Interpolation

Haoxian Zhang[1,3][0000−0001−7078−868X], Yang Zhao[2,3][0000−0002−4032−8049], and Ronggang Wang*[1,3][0000−0003−0873−0465]

[1] School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School
haoxianz@pku.edu.cn, rgwang@pkusz.edu.cn
[2] School of Computer and Information, Hefei University of Technology
yzhao@hfut.edu.cn
[3] Peng Cheng Laboratory

Abstract. Video frame interpolation (VFI) aims at synthesizing new video frames in-between existing frames to generate smoother high frame rate videos. Current methods usually use the fixed pre-trained networks to generate interpolated-frames for different resolutions and scenes. However, the fixed pre-trained networks are difficult to be tailored for a variety of cases. Inspired by classical pyramid energy minimization optical flow algorithms, this paper proposes a recurrent residual pyramid network (RRPN) for video frame interpolation. In the proposed network, different pyramid levels share the same weights and base-network, named recurrent residual layer (RRL). In RRL, residual displacements between warped images are detected to gradually refine optical flows rather than directly predict the flows or frames. Owing to the flexible recurrent residual pyramid architecture, we can customize the number of pyramid levels, and make trade-offs between calculations and quality based on the application scenarios. Moreover, occlusion masks are also generated in this recurrent residual way to solve occlusion better. Finally, a refinement network is added to enhance the details for final output with contextual and edge information. Experimental results demonstrate that the RRPN is more flexible and efficient than current VFI networks but has fewer parameters. In particular, the RRPN, which avoid over-reliance on datasets and network structures, shows superior performance for large motion cases.

Keywords: Video frame interpolation, Customizable pyramid network, Arbitrary resolution and scenes, Adjustable calculation.

1 Introduction

Video frame interpolation (VFI) is a classic computer vision task with a wide range of applications, such as novel view interpolation synthesis [9], frame rate conversion [24], slow motion [15]. As deep learning has achieved significant success in many computer vision tasks, increasingly more deep-learning-based methods are proposed to obtain high quality interpolated frames. Long *et al.* [22]

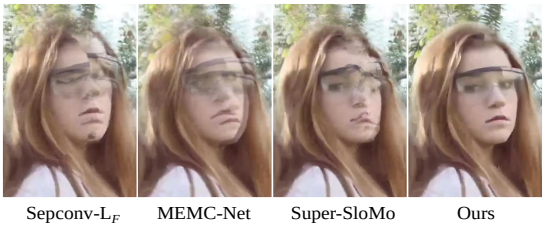


Fig. 1. A challenging **4K** (**3840** $\times$ **2160**) example from DAVIS [28]. As the resolution increases, the pixel displacements increase, which makes fixed pre-trained models difficult to estimate motion. Our method can handle arbitrary resolution cases with flexibility and adjustable calculation.

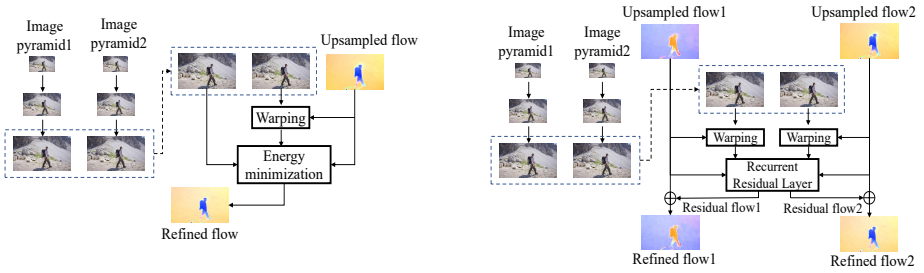


Fig. 2. ***Left***: The traditional pyramid energy minimization optical flow architecture [6, 11] with flexibility. ***Right***: The proposed recurrent residual pyramid architecture inherits the flexible structure from traditional method, which estimates residual flows between two warped images at each pyramid level by a reusable CNN.

regard VFI as an image generation task, and use CNNs to directly generate intermediate frames without an intermediate motion estimation step, which may cause blurry results. To avoid blurring artifacts and produce high quality intermediate frames, many CNN-based methods utilize an effective intermediate motion estimation step before frame interpolation [15, 21, 26, 42, 3]. However, large motion and occlusions are still challenging for these CNN-based approaches.

To handle large motion in VFI, current methods rely on improving model architecture, increasing the number of parameters, and enlarging the training set that contains enough large motion cases. For example, Van *et al.* [1], Bao *et al.* [3] and Niklaus *et al.* [25] adopt coarse-to-fine flow estimation architecture to estimate more accurate optical flow. Niklaus *et al.* [27] train a big enough spatially-adaptive convolution kernel for each pixel to cover large motion. But these fixed pre-trained models face two problems. First, as shown in Figure 1, when encountering cases with larger resolutions or motion scales that were not considered in training data, the performance of these methods tends to be significantly degraded. Second, a model originally trained for high resolution videos leads to an increased number of parameters. When applied for smaller resolution or motion scale videos, the oversized network has significant computational redundancy. Hence, fixed pre-trained model is not very effective to handle a wide

variety of scenes in the wild. Is it possible to train a flexible network that can be tailored for different scenarios, instead of training separate networks?

To address this question, we propose a flexible coarse-to-fine network inspired by classical pyramid energy minimization optical flow estimation algorithms. This architecture can customize the number of pyramid layers and make trade-offs between calculations and quality based on the application scenario. As shown in Figure 2 (*Left*), classical pyramid optical flow algorithm usually utilizes energy minimization operation to detect residual flow between first image and the warped second image at each pyramid level, which is warped by upsampled flow from last pyramid level. These algorithms can design different numbers of layers for various cases of different difficulties. Meanwhile, high quality optical flow can be accurately estimated by means of the coarse-to-fine refinement. In VFI tasks, we mainly solve optimal optical flow from the target time position to adjacent two frames, and then pre-warp frames guided by flow and blend them to final output. Therefore, the similar structure can naturally be applied to solve motion estimation for VFI with flexibility, as in Figure 2 (*Right*). This structure can arbitrarily divide a difficult large task into multiple simple small tasks, and a CNN-based module is utilized to solve the sub-problem in each pyramid layer. In this way, the optical flows would be iteratively refined by residual flows with the same network, which avoid over-reliance on datasets and network complexity.

Therefore, a flexible Recurrent Residual Pyramid Network (RRPN) is proposed in this paper. In the training phase, we design a multi-layers recurrent pyramid network, in which each pyramid layer shares same structure and weights. As mentioned before, the residual learning strategy is used in the pyramid network to gradually refine the flow. Hence, each pyramid layer is named as recurrent residual layer (RRL). Occlusion maps, which are also generated in this recurrent and residual way, are applied to the warped images before fusion to solve occlusion better. In the testing phase, the RRL can be easily applied more times than that in training phase. Moreover, in order to improve the details and sharpness of the output frame, a refinement network is presented after the pyramid, which simultaneously takes the warped frames, warped contextual feature, warped edge maps and occlusion maps as input. The contextual feature and the edge maps are extracted via pre-trained VGG19 [33] and HED [40].

The entire network is trained end-to-end using more than 80K collected frame groups. Experimental results demonstrate that the recurrent residual approach can achieve state-of-the-art performance on several datasets, including Middlebury [2], UCF101 [34], Thumos15 [16] (720P videos), ActivityNet [8] (1080P videos) and H.266 4K test sequences [5, 35], with higher flexibility and lower complexity. In particular, the proposed method shows superior performance when facing with large motion cases that not contained in the training data.

2 Related Work

Common frame interpolation approaches usually generate intermediate frames with an intermediate motion estimation step [21], which usually is optical flow [29,

36, 12]. In this section, we mainly discuss recent learning-based VFI and optical flow estimation algorithms.

Video frame interpolation. Deep voxel flow [21] estimates a 3D optical flow across space and time, to samples pixels from adjacent frames. However, inaccuracies in voxel flow estimation lead to unsatisfactory results sometimes. Van *et al.* [1] combine DVF with coarse-to-fine architecture to achieve better results, while its performance is still limited by the number of the pyramid levels. Niklaus *et al.* [27] utilize a CNN to combine motion estimation and pixel synthesis into a single convolution step. They estimate spatially-adaptive convolution kernels for each pixel to synthesize a intermediate frame. While their prediction are limited by the size of adaptive kernels when faced with large motion. Bao *et al.* [4] first warp input frames by optical flow and then estimate kernels to sample pixels, which inherits the benefits from both flow-based and kernel-based approaches. Jiang *et al.* [15] estimate bidirectional flow between two frames, and then synthesize intermediate flow fields to generate the intermediate frame at the arbitrary time step . Liu *et al.* [20] propose a cycle consistency loss to make synthesized frames more reliable by reconstructing input frames with synthesized frames. Recently, Zhang *et al.* [43] effectively uses spatio-temporal information contained in multiple frames to generate high-quality intermediate frames.

Optical flow estimation. As a pioneer of CNN-based methods, Dosovitskiy *et al.* [7] develop two network architectures of FlowNetS and FlowNetC, which proved that a U-Net [32] architecture can be used to predict optical flow effectively. Ilg *et al.* [13] design a much larger FlowNet2 based on FlowNetS and FlowNetC to achieve better performance. In addition to the supervised learning, learning optical flow using CNNs in an unsupervised way has also been explored [19, 23, 30, 37]. Recently, many deep networks are designed by considering classical principles of optical flow, such as coarse-to-fine strategy and iterative residual refinement, and have achieved better results with less computation [12, 29, 36].

3 Proposed Approach

Given consecutive two input frames $I = \{I_0, I_1\}$, the goal of VFI is to predict the intermediate frame I_t at the temporal location t in between I_0 and I_1, $t \in (0, 1)$. Let us assume $F = \{F_{t\to0}, F_{t\to1}\}$ to represent the predicted optical flow from I_t to I_0 and I_1. The intermediate frame I_t can be synthesized through warping two frames guided by these flow and then fusing them as follows:

$$I_t = M_{t\leftarrow0} \otimes w(I_0, F_{t\to0}) + M_{t\leftarrow1} \otimes w(I_1, F_{t\to1}), \tag{1}$$

where $w(\cdot, \cdot)$ denotes a backward warping function, which can be implemented using bilinear interpolation [21] and is differentiable. $M = \{M_{t\leftarrow0}, M_{t\leftarrow1}\}$ denote occlusion maps of the two warped frames, where $\sum_{i=0}^{1} M_{t\leftarrow i}(i, j) = 1$, $M_{t\leftarrow i}(i, j) \in [0, 1]$. (i, j) denote the pixel coordinate and $\otimes$ denotes element-wise multiplication. Occlusion areas often results in artifacts in the warped frames. Therefore,

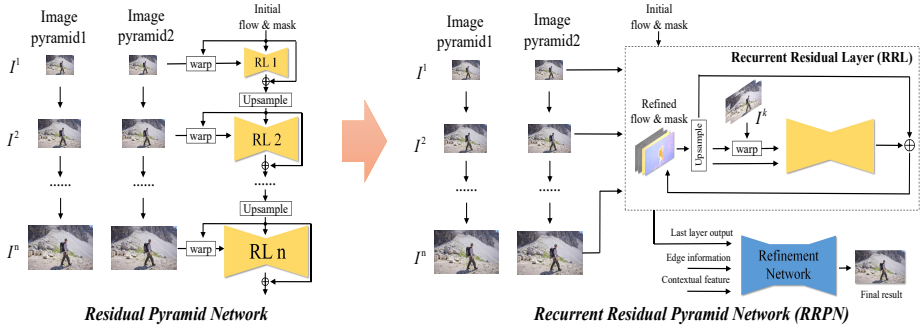


Fig. 3. ***Left***: A Residual Pyramid Network with several residual layers (RL) to detect residual flows between warped images at each pyramid level. ***Right***: Overview of the Recurrent Residual Pyramid Network (RRPN), which utilizes the single recurrent residual layer (RRL) with shared weights at each pyramid level to iteratively update optical flows. Moreover, a refinement network that combines edge information and contexture feature are used to enhance the final output.

occlusion masks [15, 42] are estimated and only pixels that are not occluded are used in interpolation.

3.1 Recurrent Residual Pyramid Network (RRPN)

Pyramid framework is commonly used in traditional computer vision and pattern recognition tasks, which can effectively divide a difficult task into multiple simple tasks, especially for motion estimation. Residual learning strategy is also useful in many CNN-based image restoration methods that utilize a global residual connection to improve the convergency and force the networks to learn the high-frequency details. To inherit the benefits of these effective strategies, we first present a residual pyramid network similar to FIGAN [1], in which a series of base-networks are composed in coarse-to-fine manner to refine VFI results, as shown in Figure 3 (*Left*). In each pyramid layer, residual displacement are predicted from warped images, and then propagate to higher resolution layers of the pyramid to update optical flows. Hence, the optical flows are gradually improved until high-quality optical flows are obtained at full resolution. In order to avoid error propagation in the iterative process, all warped images are resampled by updated optical flows from original input at each pyramid level, rather than being resampled by residual flows from warped images. Therefore, the unsupervised motion information are kept track of, composed, and passed through the network instead of being absorbed into warped images. By refining optical flows and occlusion masks with residual flows and residual masks at each pyramid level, the estimation accuracy of motion and occlusion can also be increased. Note that the values of initial flows and masks are 0 and 0.5, respectively.

However, this fixed residual pyramid network still cannot well handle a wide variety of VFI scenes in the wild, because different numbers of pyramid layers should be set for videos with different resolutions. Moreover, increasing the

layers of residual pyramid network also lead to larger amount of parameters. Hence, design of numbers of pyramid layers becomes another difficult question. One common way is to carefully select the number of layers to make a balance between performance and computational complexity. However, a pyramid with fixed layers either is insufficient to deal with complex larger motion cases, or increases computational redundancy for easy smaller motion cases. In order to address this problem in a flexible and efficient way, we propose a Recurrent Residual Pyramid Network (RRPN) based on weight sharing strategy. Each layer of the pyramid uses the same network with shared weights to detect the residual displacement of warped images. Therefore, the number of pyramid layers can be customized according to the application scenario to achieve a trade-off between calculations and quality.

The structure of proposed RRPN is shown in Figure 3 (*Right*), which adopts similar architecture as the residual pyramid network. However, each pyramid layer adopts the same base-network with shared weights in recurrent way, named Recurrent Residual Layer (RRL). Finally, a refinement network is presented to further enhance the details of final output with contextual and edge information. Let $u(\cdot)$ be the upsampling function using bilinear interpolation. I^k denotes the image from k-th layer of the image pyramid. β^k denotes the ratio of the resolution between I^k and I^{k-1}. M^k, m^k, F^k and f^k denote occlusion mask M, residual mask m, optical flow F and residual flow f, respectively. At the k-th level of the pyramid, F^k and M^k can be described as follows,

$$f^k, m^k = RRL(w(I^k, u(\beta^k F^{k-1})), u(\beta^k F^{k-1}), u(M^{k-1})) \tag{2}$$

$$F^k = u(\beta^k F^{k-1}) + f^k \tag{3}$$

$$M^k = u(M^{k-1}) + m^k \tag{4}$$

3.2 Recurrent Residual Layer (RRL)

As shown in Figure 4, RRL estimates the residual displacements between warped images to gradually refine optical flows rather than directly predicts the flows or frames. The backbone of RRL is a U-Net architecture. (The configuration details are provided in the supplementary material.) Moreover, the feature extractor consists of 3 convolutional layers and the context network is design based on dilated convolutions, which has 4 convolutional layers with dilation constants of [2, 4, 8, 1]. The spatial kernel for all convolutional layers above is 3×3 except the first hierarchy of U-Net encoder, which adopts 7×7 kernels.

In this paper, we train a 3-level RRPN with the same RRL at each pyramid level to enforce the RRL to learn residual displacement detection. This results in a single effective unsupervised optical flow predictor RRL that can be applied multiple times across pyramid structure. The predicted flows are visualized in Figure 5. We can observe that residual flows and residual masks can be effectively predicted to refined the optical flow and occlusion masks in coarse-to-fine way, in spite of using the same RRL at each pyramid layer. Note that the RRL can be easily applied more times in testing phase than that in training stage,

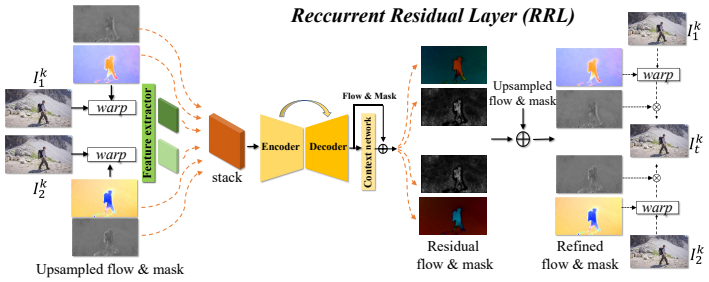


Fig. 4. Illustration of the RRL and only the target frame at each pyramid level is used as supervisory signal. The RRL uses the same siamese convolutional layer as feature extractor to provide good features to establish correspondence, particularly in the presence of shadows and lighting changes. Moreover, context network is also used to post-process the residual flow and mask[36].

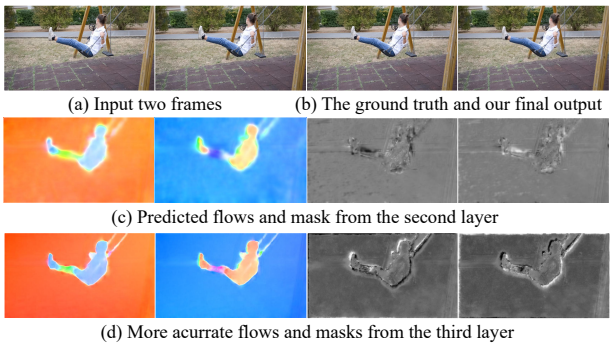


Fig. 5. Samples of predicted flows and masks from a 3-level Recurrent Residual Pyramid Network that indicates the single RRL can gradually refine the result.

and the performance continues to increase until saturation. This implies that RRL can help to achieve better performance with flexibility. In addition, this compact and flexible method also sheds the reliance on large motion datasets. By inheriting the merits of traditional pyramid framework, the RRPN can arbitrarily decompose large prediction task into multiple simple small prediction tasks. Therefore, the RRL can be only trained to learn small increment motion estimation and does not require a dataset which covers a wide range of motion.

3.3 Refinement Network

To further enhance the visual quality of output frame, a refinement network, which consists of 3 residual blocks, is added after the last layer of the recurrent residual pyramid to predict the residuals between the ground-truth frame and the blended frame. Generating the final output via blending two warped frames with occlusion maps usually leads to the loss of rich contextual information [25]. Meanwhile pixels with larger gradients tend to have large errors [20]. Therefore,

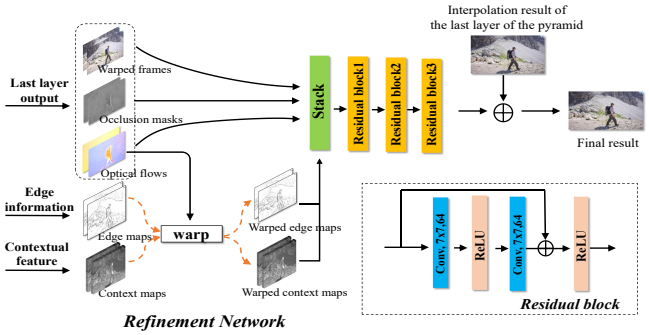


Fig. 6. Illustration of the refinement network.

we use *conv1* layer of a pre-trained VGG19 [33] and *side-output1* of HED [40] to extract the contextual feature and the edge map of original frames. Then we concatenate the warped input frames, occlusion maps, output flows, warped contextual feature, and warped edge maps as input, as shown in Figure 6.

3.4 Loss Function

For each pyramid layer of d-level RRPN, we denote the interpolated frame by $\widehat{I}_t^k$ and its ground truth by I_t^k. Moreover, $\widehat{I}_t$ and I_t denote the output of refinement network and the target frame, respectively. We mainly use *Reconstruction loss* and *Edge-aware smoothness loss* in this paper.

Reconstruction loss l_{r1} and l_{r2} [15, 21, 3] are traditional MAE loss functions, where pixel values are normalized into the range [-1, 1].

$$l_{r1} = \sum_{k=1}^{d} \left\| I_t^k - \hat{I}_t^k \right\|_1 \tag{5}$$

$$l_{r2} = \left\| I_t - \hat{I}_t \right\|_1 \tag{6}$$

Edge-aware smoothness loss l_s [10], which is a spatial coherence regularization, is added to encourage neighboring pixels to have similar flow values. As flow discontinuities often occur at image gradients, we weight this cost with an edge-aware term by means of image gradients, where N^k is the number of pixels at each pyramid level and (i, j) denote the pixel coordinate.

$$l_s = \sum_{k=1}^{d} \frac{1}{N^k} \sum_{i,j} \left\| \partial_x F_{ij}^k \right\| e^{-\left\| \partial_x I_{t,ij}^k \right\|} + \left\| \partial_y F_{ij}^k \right\| e^{-\left\| \partial_y I_{t,ij}^k \right\|} \tag{7}$$

Finally, the loss function l_{RRL} and $l_{refinement}$ are defined as follow, the parameters are empirically set as $\lambda_r = 1$, $\lambda_s = 0.01$, $d = 3$.

$$l_{RRL} = \lambda_r l_{r1} + \lambda_s l_s \tag{8}$$

$$l_{refinement} = l_{r2} \tag{9}$$

4 Experiments

4.1 Training

Training Dataset. For training, we collect 60-fps videos with a resolution of 1280×720 from YouTube, which contain a great variety of scenes. Then videos are split into triplets of three frames and all frames are resized to have a shortest dimension of 480. For each triplet, the middle frame serves as the ground truth while the other two are inputs. To have more challenging samples for training, we only select triplets with useful information, especially large motion. Hence we calculate optical flow between input frames using DIS flow [18] to drop samples with no or little motion. Finally, approximately 80,000 triplets are selected, 4000 samples are used for validation and 4000 samples are used as testing data for model analysis among them. We augment the training data by randomly cropping patches with a size of 352×352, flip each patch vertically or horizontally, and swap the temporal order.

Implementation Details. We pre-train the RRL and refinement network in turn, and then fine-tune the entire model. Adam [17] is used to optimize the proposed network. We set the β_1 and β_2 to 0.9 and 0.999 and use a batch size of 8. The learning rate is initialized to be $1e-4$, $1e-5$ for pre-train stage and fine-tune stage respectively, and decreased by a factor of 10 every 15 epochs. Batch normalization [14] is adopted on RRL for accelerating convergence. We train our network to interpolate intermediate frame at $t = 0.5$ temporal location in all experiments. Moreover, we train our model on an NVIDIA Tesla V100 GPU, which takes about one day to converge.

4.2 Evaluation Datasets and Metrics

The proposed method is evaluated on on several independent datasets with different resolutions, including UCF101 (240P), Middlebury benchmark (480P), our testing data (480P), Thumos15 (720P), ActivityNet (1080P) and H.266 4K test sequences [5, 35].

UCF101 (240P). Videos from UCF101 are low resolution and relatively easy to interpolate intermediate frames. So we select videos with obvious motion using DIS flow, which are more difficult than that used in DVF [21].

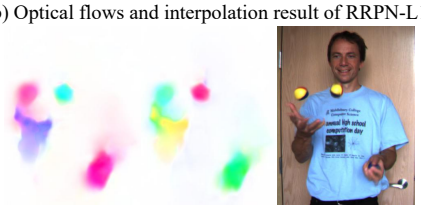
Middlebury benchmark (480P). Since the interpolation category of the Middlebury optical flow benchmark is typically used for assessing frame interpolation methods, we submit our frame interpolation results to its website.

Thumos15 (720P), ActivityNet (1080P) and H.266 test sequences (4K). To verify the performance of our approach in high-resolution videos, we select 25 720P videos from Thumos15 test data, 20 1080P videos from ActivityNet data and all 6 4K (3840×2160) test sequences of VVC (H.266) video codec standard. These high-resolution videos contain a variety of situations, such as small and large movement, motion blur, global motion, and occlusion.

Metrics. PSNR, SSIM [38] and the interpolation error (IE) [2], which is defined as root-mean-square difference between the ground-truth and the pre-

(a) Input two frames and the ground truth (red box)

(b) Optical flows and interpolation result of RRPN-L1



(c) Optical flows and interpolation result of RRPN-L3

Fig. 7. Examples for the effectiveness of customizing the number of pyramid levels.

Table 1. Impact of the number of pyramid layer.

		PSNR	SSIM	IE
UCF101(240P)	RRPN-L1 w/o R	34.64	0.960	6.06
	RRPN-L2 w/o R	**34.72**	**0.962**	**5.98**
Our test set(480P)	RRPN-L1 w/o R	30.74	0.878	8.62
	RRPN-L2 w/o R	31.75	**0.913**	7.95
	RRPN-L3 w/o R	**31.80**	**0.913**	**7.92**
Thumos15(720P)	RRPN-L1 w/o R	33.56	0.936	7.89
	RRPN-L2 w/o R	34.50	0.946	7.04
	RRPN-L3 w/o R	34.65	**0.951**	6.83
	RRPN-L4 w/o R	**34.69**	**0.951**	**6.78**

Table 2. Effectiveness of the refinement network (test on 480P).

	PSNR	SSIM	IE
RRPN-L2 w/o R	31.75	0.913	7.95
RRPN-L2 (basic R)	31.78	0.910	7.93
RRPN-L2 (edge R)	31.83	0.913	7.89
RRPN-L2 (context R)	31.86	0.915	**7.85**
RRPN-L2 (whole R)	**31.90**	**0.916**	7.86

diction, are used to evaluate the quality of interpolated video frame. Lower IE indicate better performance.

4.3 Model Analysis

We analyze the contribution of the two key components in the proposed model: recurrent residual pyramid architecture and refinement network. UCF101 (240P), our testing data (480P), and Thumos15 (720P) are used here.

Customizing the number of pyramid layers for the RRPN. To analyze the flexibility and effectiveness of the RRPN, we customize a series of RRPNs with different number of pyramid layers and then evaluate their performance on several datasets with various resolutions. Note that refinement network is not used for all these models in this testing. Table 1 shows the impact of the number of pyramid layers on interpolation performance. '-Lx' indicates the number of pyramid layers.

We can see that the quality of the interpolation can be improved by means of more pyramid layers, although each layer of the pyramid is with same network and weights. Moreover, when facing with large resolution (large motion) cases on Thumos15 (720p) dataset, using four RRLs can continue to increase the performance although this RRL is originally trained on a 3-level RRPN. These verify

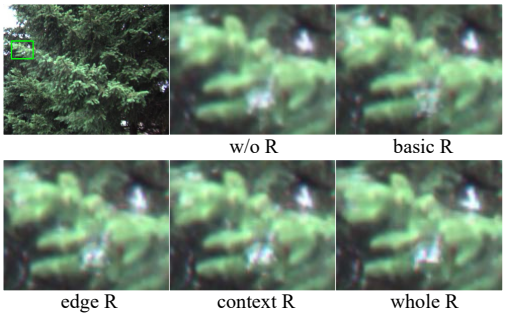


Fig. 8. Examples ('Evergreen' on the Middlebury set) for the effectiveness of refinement network. The context and edge maps can help produce sharper results in the highly textured region.

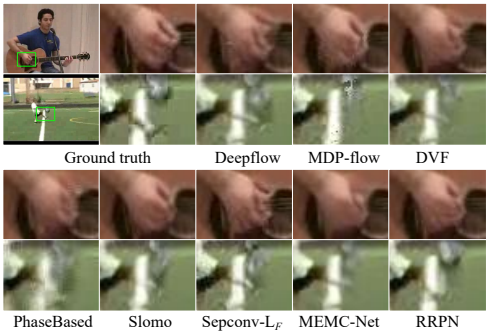


Fig. 9. Challenging sample results from our selected UCF101 dataset.

	PSNR	SSIM	IE
MDP-Flow2 [41]	34.49	0.959	6.37
DeepFlow [39]	34.40	0.957	6.44
Phase-Based [24]	33.65	0.946	6.83
SepConv-L_F [27]	34.62	0.959	6.30
DVF [21]	34.13	0.956	6.35
Slomo [15]	34.59	0.960	6.06
DAIN [3]	34.75	**0.963**	**5.93**
MEMC-Net [4]	34.70	**0.963**	5.95
RRPN-L1 (Match for 240P)	34.73	0.962	5.95
RRPN-L2	**34.76**	0.962	**5.93**

	DVF [21]	ToFlow [42]	DAIN [3]	RRPN-L1
PSNR	34.12	34.58	34.99	34.76

Table 3. Comparison on our UCF101 dataset (*Top*) and the UCF101 dataset (256×256) used in DVF [21] (*Bottom*).

the core idea of the RRPN and indicate our approach can can flexibly deal with large motion. Figure 7 provides a visualization of optical flows and interpolated frames for large motion cases. We can observe that the single RRL can only detect small displacements, so there are obvious artifacts in the hand and ball with large motion in Figure 7(b). But it can divide large displacement prediction into multiple simple small displacement predictions in pyramid recurrent way to gradually capture large motion, as shown in Figure 7(c).

However, the performance improvement brought by increasing the numbers of pyramid layers would gradually reaches saturation. As the Table 1 shows, one layer is sufficient for 240P frames, but the 480P and 720P frames may require 2 and 3 layers, respectively. Therefore, the flexibility of the RRPN is not only reflected in large motion cases, but also in making trade-offs between calculations and quality for cases of different difficulty levels.

Impact of the refinement network. In this part, four variants of refinement networks: whole refinement network (whole R), refinement network with-

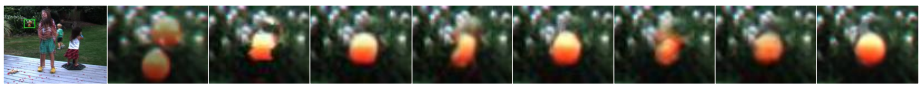


Fig. 10. Visual comparisons on the Middlebury benchmark. The proposed method reconstructs a clear shape of the ball.

Table 4. Evaluation on the Middlebury(480P). The RRPN-L2 has comparable performance with DAIN in terms of IE and NIE but with fewer parameters and calculation.

	AVERAGE		Mequon		Schefflera		Urban		Teddy		Backyard		Basketball		Dumptruck		Evergreen	
	IE	NIE	IE	NIE	IE	NIE	IE	NIE	IE	NIE	IE	NIE	IE	NIE	IE	NIE	IE	NIE
SepConv [27]	5.61	0.83	2.52	0.54	3.56	0.67	4.17	1.07	5.41	1.03	10.2	0.99	5.47	0.96	6.88	0.68	6.63	0.70
ToFlow [42]	5.49	0.84	2.54	0.55	3.70	0.72	3.43	0.92	5.05	0.96	9.84	0.97	5.34	0.98	6.88	0.72	7.14	0.90
Slomo [15]	5.31	0.78	2.51	0.59	3.66	0.72	2.91	0.74	5.05	0.98	9.56	0.94	5.37	0.96	6.69	0.60	6.73	0.69
CtxSyn [25]	5.28	0.82	2.24	0.50	2.96	0.55	4.32	1.42	4.21	0.87	9.59	0.95	5.22	0.94	7.02	0.68	6.66	0.67
MEMC-Net [4]	5.24	0.83	2.47	0.60	3.49	0.65	4.63	1.42	4.94	0.88	8.91	0.93	4.70	0.86	6.46	0.66	6.35	0.64
DAIN [3]	4.86	0.71	2.38	0.58	3.28	0.60	3.32	0.69	4.65	0.86	7.88	0.87	4.73	0.85	6.36	0.59	6.25	0.66
RRPN-L2	4.93	0.75	2.38	0.53	3.70	0.69	3.29	0.87	5.05	0.94	8.20	0.88	4.38	0.88	6.50	0.65	6.00	0.62

out context and edge maps (basic R), refinement network without context maps (edge R) and refinement network without edge maps (context R), are added behind RRPN-L2 to test our 480P test set. The quantitative results and interpolated images are shown in Table 2 and Figure 8, which demonstrate context and edge maps can improve the performance and reproduce sharper results.

4.4 Comparison with State-of-the-art Methods

In this section, our approach is compared with state-of-the-art methods published on Middlebury benchmark, including MDP-Flow2 [41], DeepFlow [39], Phase based approach from [24], SepConv [27], DVF [21], recent Super-Slomo [15], MEMC-Net [4] and DAIN [3]. For all these methods, we use the source code or pre-trained models from the original papers. For optical flow methods, we apply the interpolation algorithm presented in [2]. UCF101 (240P), Middlebury (480P), Thumos15 (720P), ActivityNet (1080P) and H.266 test sequences (4K) are adopted for evaluation here.

UCF101. In this part, we utilize RRPN with one layer (RRPN-L1) to compare with other state-of-the-art methods on our UCF101 dataset. The quantitative results are shown in Table 3. RRPN-L1 has comparable performance and outperforms most methods in low resolution cases with fewer parameters. Moreover, the performance of RRPN-L1 on the UCF101 dataset (256×256) used in DVF [21] is consistent with results of our UCF101 dataset, while samples in our UCF101 dataset have larger motion. Sample interpolation results from our UCF101 can be found at Figure 9.

Middlebury. The image resolution in Middlebury is around 640×480 pixels. Therefore, we use RRPN with just two layers (RRPN-L2) to test eight sequences provided by the Middlebury benchmark, and submit our frame interpolation results to its website. Normalized Interpolation Error (NIE) is also used on the Middlebury. In Table 4, we show the comparisons on the EVALUATION

Table 5. Results on the Thumos15 720P (*Left*) and ActivityNet 1080P (*Right*).

	PSNR	SSIM	IE
DeepFlow [39]	33.65	0.946	7.67
Phase-Based [24]	32.77	0.927	8.42
SepConv-L_F [27]	33.73	0.940	7.79
DVF [21]	33.46	0.937	8.03
Slomo [15]	33.81	0.943	7.98
MEMC-Net [4]	33.96	0.948	6.99
DAIN [3]	34.53	0.950	7.02
RRPN-L3	**34.77**	**0.951**	**6.73**

	PSNR	SSIM	IE
SepConv-L_F [27]	28.86	0.883	12.00
DVF [21]	28.88	0.874	12.98
Slomo [15]	29.04	0.891	11.71
MEMC-Net [4]	29.20	0.890	11.57
DAIN [3]	29.29	0.892	11.44
RRPN-L3	**30.13**	0.898	10.93
RRPN-L4	30.08	**0.902**	**10.68**

Table 6. Results on the H.266(VVC) 4K (3840×2160) test sequences.

	PSNR	SSIM	IE
SepConv-L_F [27]	32.74	0.939	7.48
DVF [21]	32.81	0.937	7.41
Slomo [15]	33.54	0.948	6.98
MEMC-Net [4]	33.62	0.947	6.94
RRPN-L4	34.86	0.960	6.25
RRPN-L5 (Match for the resolution)	**35.28**	**0.961**	**6.11**

Table 7. Comparisons on parameter and runtime (test on 480P).

	#Parameters (million)	Runtime (seconds)
SepConv [27]	21.6	0.15
MEMC-net [4]	70.3	0.10
Slomo [15]	39.6	0.14
RRPN-L2	6.1	0.12
RRPN-L2 w/o R	4.8	0.06
RRPN-L3	6.1	0.12
RRPN-L3 w/o R	4.8	0.07

set of the benchmark. The proposed model not only outperforms representative non-neural methods based on optical flow, such as Deep flow, MDP-flow2, Epicflow [31], but also performs favorably against recent CNN-based approaches, like CtxSyn [25], ToFlow [42], Slomo, Sepconv, MEMC-Net. Our network with just two pyramid layers here balance calculations and quality well, which has comparable performance with DAIN [3] but with fewer parameters. Sample interpolation results from 'Backyard' sequences are shown Figure 10.

High resolution videos. For Thumos15 (720P), ActivityNet (1080P) and H.266 (4K) test data, we use RRPN with three layers (RRPN-L3), four layers (RRPN-L4) and five layers (RRPN-L5) to interpolate intermediate frames, respectively. As reported in Table 5 and 6, our approach can achieve superior performance in higher resolution videos, which reflects the advantages of custom pyramid layers in dealing with large motion. Qualitative comparisons are shown in Figure 11 and 12. For super large motion cases that have not included in training data, the RRPN can produce visually pleasing results with fewer artifacts, while other methods tend to produce significant artifacts.

Computational efficiency. We list the number of model parameters and execution time (640×480 image, a Tesla V100 GPU) of each method in Table 7. Please see supplementary material for more network details. Compared with representative state-of-the-art methods, the proposed model is more compact and run faster. The RRPN-L2 has 71% fewer parameters than SepConv and save 20% execution time. Morover, the RRPN-L2 w/o refinement network can further save 21% parameters and 50% runtime.

5 Conclusion

Motivated by classical pyramid energy minimization optical flow algorithm, this paper proposed a compact and flexible network to handle large motion for video

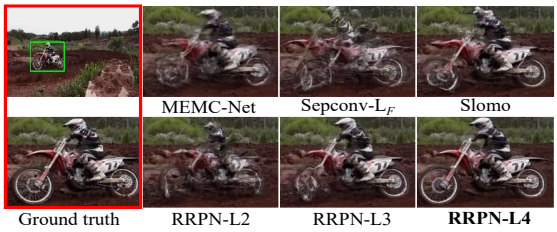


Fig. 11. Sample interpolation results from ActivityNet (**1080P**) videos. The proposed method can better restore the shape of the motorcycle, which is an challenging example with large motion.

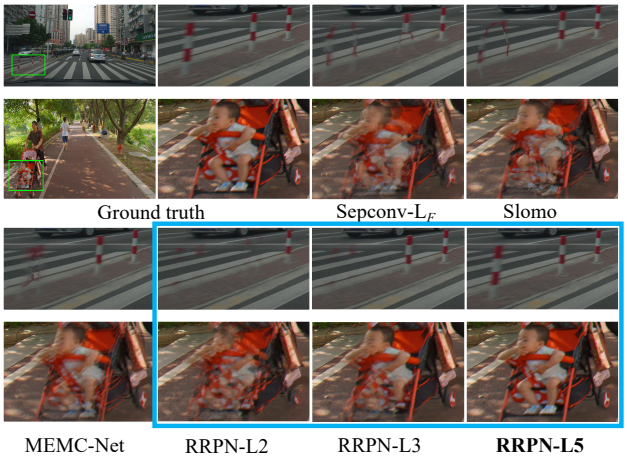


Fig. 12. Sample interpolation results from H.266 (**4K**) test data. Our method can gradually capture large motion for the 4K video examples, in which the pole and the baby carriage closer to the camera have larger motion. While other approaches produce significant artifacts on these super large motion cases that are not considered. Please see supplementary material for more image and video comparison.

frame interpolation, named Recurrent Residual Pyramid Network (RRPN). The proposed RRPN adopts the same Recurrent Residual Layer (RRL) with shared weights at each pyramid layer to predict residual flows in between warped images. Therefore, the RRPN can customize the number of pyramid layers according to different video resolutions and thus make trade-offs between complexity and visual quality. Moreover, a refinement network is introduced to further enhancing details of the interpolated frame. Experiments demonstrate that the RRPN is more flexible and efficient than current SOTA methods but has fewer parameters.

Acknowledgement: Thanks to National Natural Science Foundation of China 61672063 and 61972129, Shenzhen Research Projects of JCYJ20180503182128089 and 2018060-80921419290.

References

1. van Amersfoort, J., Shi, W., Acosta, A., Massa, F., Totz, J., Wang, Z., Caballero, J.: Frame interpolation with multi-scale deep loss functions and generative adversarial networks. arXiv preprint arXiv:1711.06045 (2017)
2. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision **92**(1), 1–31 (2011)
3. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3703–3712 (2019)
4. Bao, W., Lai, W.S., Zhang, X., Gao, Z., Yang, M.H.: Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. arXiv preprint arXiv:1810.08768 (2019)
5. Bross, B., Chen, J., Liu, S.: Versatile video coding (draft 2). In: JVET-J1001 (2018)
6. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: European conference on computer vision. pp. 25–36. Springer (2004)
7. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2758–2766 (2015)
8. Fabian Caba Heilbron, Victor Escorcia, B.G., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 961–970 (2015)
9. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: Deepstereo: Learning to predict new views from the world's imagery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5515–5524 (2016)
10. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 270–279 (2017)
11. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1-3), 185–203 (1981)
12. Hur, J., Roth, S.: Iterative residual refinement for joint optical flow and occlusion estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5754–5763 (2019)
13. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE conference on computer vision and pattern recognition (CVPR). vol. 2, p. 6 (2017)
14. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
15. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. arXiv preprint arXiv:1712.00080 (2018)
16. Jiang, Y., Liu, J., Zamir, A.R., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: Thumos challenge: Action recognition with a large number of classes (2014)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Kroeger, T., Timofte, R., Dai, D., Van Gool, L.: Fast optical flow using dense inverse search. In: European Conference on Computer Vision. pp. 471–488. Springer (2016)

19. Liu, P., Lyu, M.R., King, I., Xu, J.: Selflow: Self-supervised learning of optical flow. In: CVPR (2019)
20. Liu, Y.L., Liao, Y.T., Lin, Y.Y., Chuang, Y.Y.: Deep video frame interpolation using cyclic frame generation. In: AAAI Conference on Artificial Intelligence (2019)
21. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: ICCV. pp. 4473–4481 (2017)
22. Long, G., Kneip, L., Alvarez, J.M., Li, H., Zhang, X., Yu, Q.: Learning image matching by simply watching video. In: European Conference on Computer Vision. pp. 434–450. Springer (2016)
23. Meister, S., Hur, J., Roth, S.: Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
24. Meyer, S., Wang, O., Zimmer, H., Grosse, M., Sorkine-Hornung, A.: Phase-based frame interpolation for video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1410–1418 (2015)
25. Niklaus, S., Liu, F.: Context-aware synthesis for video frame interpolation. arXiv preprint arXiv:1803.10967 (2018)
26. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive convolution. In: IEEE Conference on Computer Vision and Pattern Recognition. vol. 1, p. 3 (2017)
27. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive separable convolution. arXiv preprint arXiv:1708.01692 (2017)
28. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Computer Vision and Pattern Recognition (2016)
29. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, p. 2. IEEE (2017)
30. Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., Zha, H.: Unsupervised deep learning for optical flow estimation. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
31. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Epicflow: Edge-preserving interpolation of correspondences for optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1164–1172 (2015)
32. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
34. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
35. Suehring, K., Li, X.: Jvet common test conditions and software reference configurations. JVET-B1010 (2016)
36. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8934–8943 (2018)
37. Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., Xu, W.: Occlusion aware unsupervised learning of optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4884–4893 (2018)

38. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
39. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: Large displacement optical flow with deep matching. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1385–1392 (2013)
40. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision. pp. 1395–1403 (2015)
41. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(9), 1744–1757 (2012)
42. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. International Journal of Computer Vision **127**(8), 1106–1125 (2019)
43. Zhang, H., Wang, R., Zhao, Y.: Multi-frame pyramid refinement network for video frame interpolation. IEEE Access **7**, 130610–130621 (2019)