AI视频插帧在实时互动场景中的应用

周世付

agora.io

目录

实时互动场景用户体验

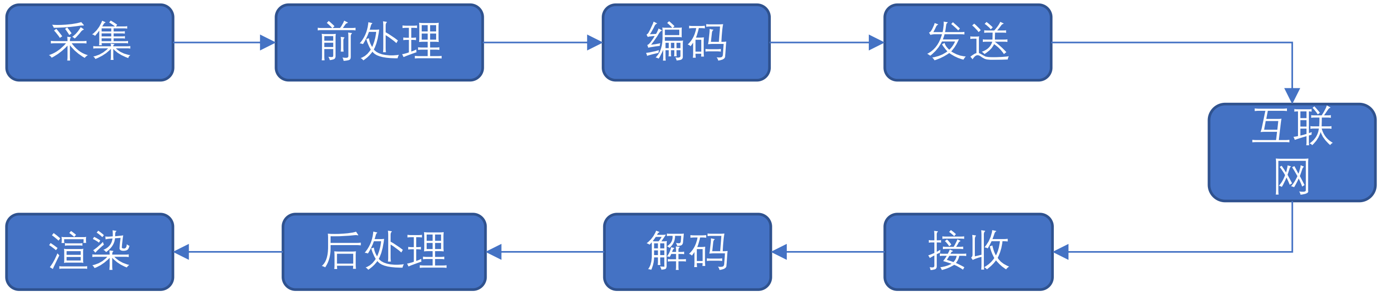
■ 低延时

声音、视频流畅

■ 高质量

声音保真、画面清晰

实时互动视频传输



- 前处理
  检测、分割、美颜、ROI
- 编解码
  pvc、roi
- 后处理
  去噪、锐化、超分、插帧

视频插帧与低延时

■ 低帧率生成高帧率

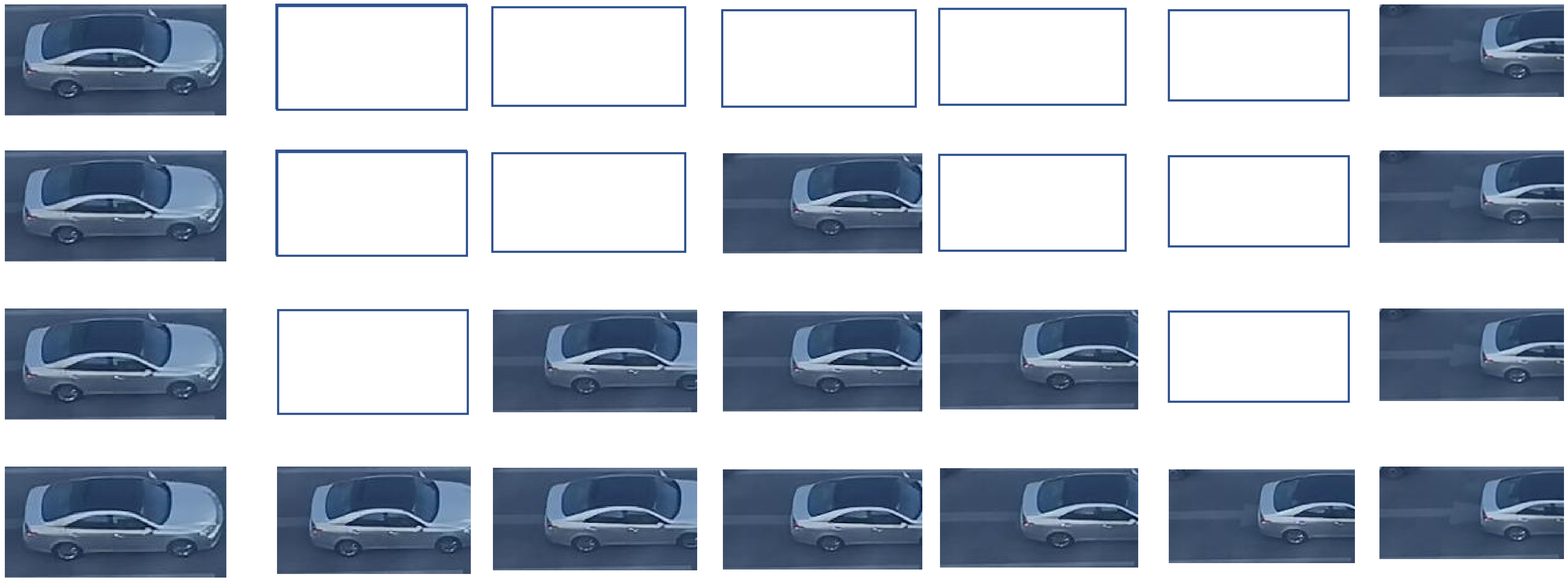
低帧率视频传输，减轻网络带宽压力，降低传输时延，接收端插帧恢复高帧率视频

■ 恢复丢失帧率

传输过程中，出现丢包，整帧数据丢弃，再重传，传输时延大； 利用前后帧，恢复中间帧，无需重传

视频插帧方法

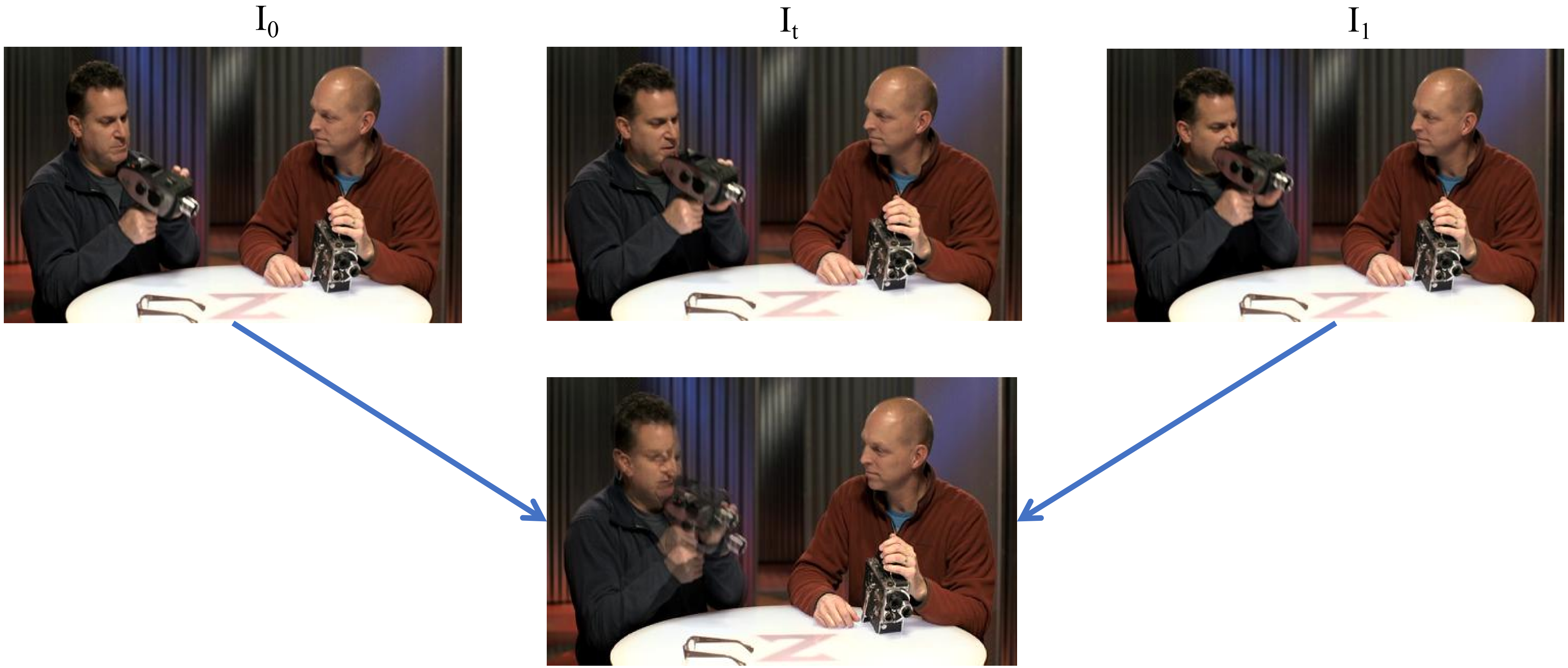
在连续的两帧图像之间，生成1帧或多帧图像



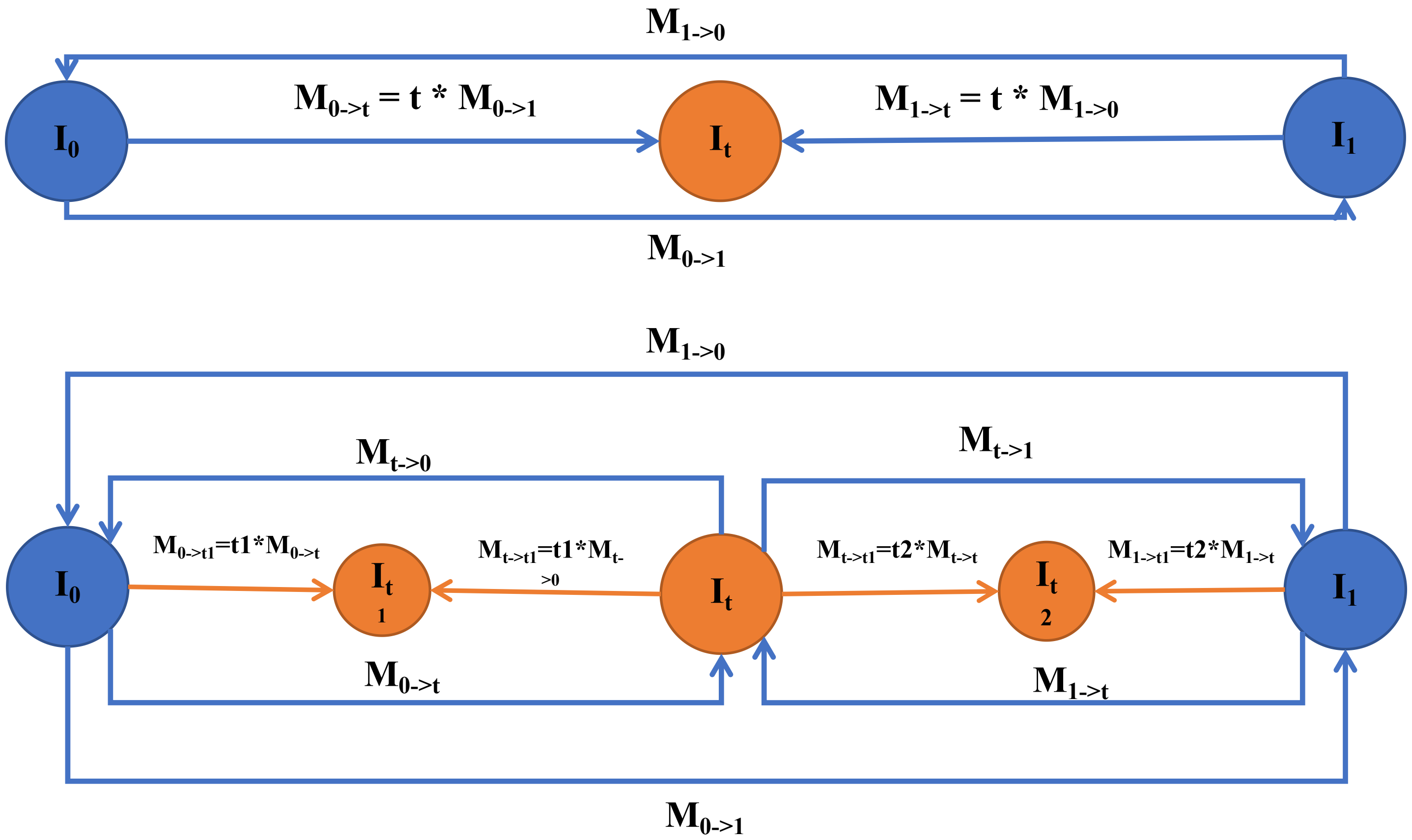
视频插帧的原理

$$I_t = \alpha * I_0 + (1-\alpha) * I_1$$

$$I_t = \alpha * g(I_0, M_{0\to t}) + (1-\alpha) * g(I_1, M_{1\to t})$$



单帧与多帧插值



M1->0
M0->t = t * M0->1
M1->t = t * M1->0
I0
It
I1
M0->1
M1->0
Mt->0
Mt->1
M0->t1=t1*M0->t
Mt->t1=t1*Mt->0
Mt->t1=t2*Mt->t
M1->t1=t2*M1->t
I0
It1
It
It2
I1
M0->t
M1->t
M0->1

运动估计

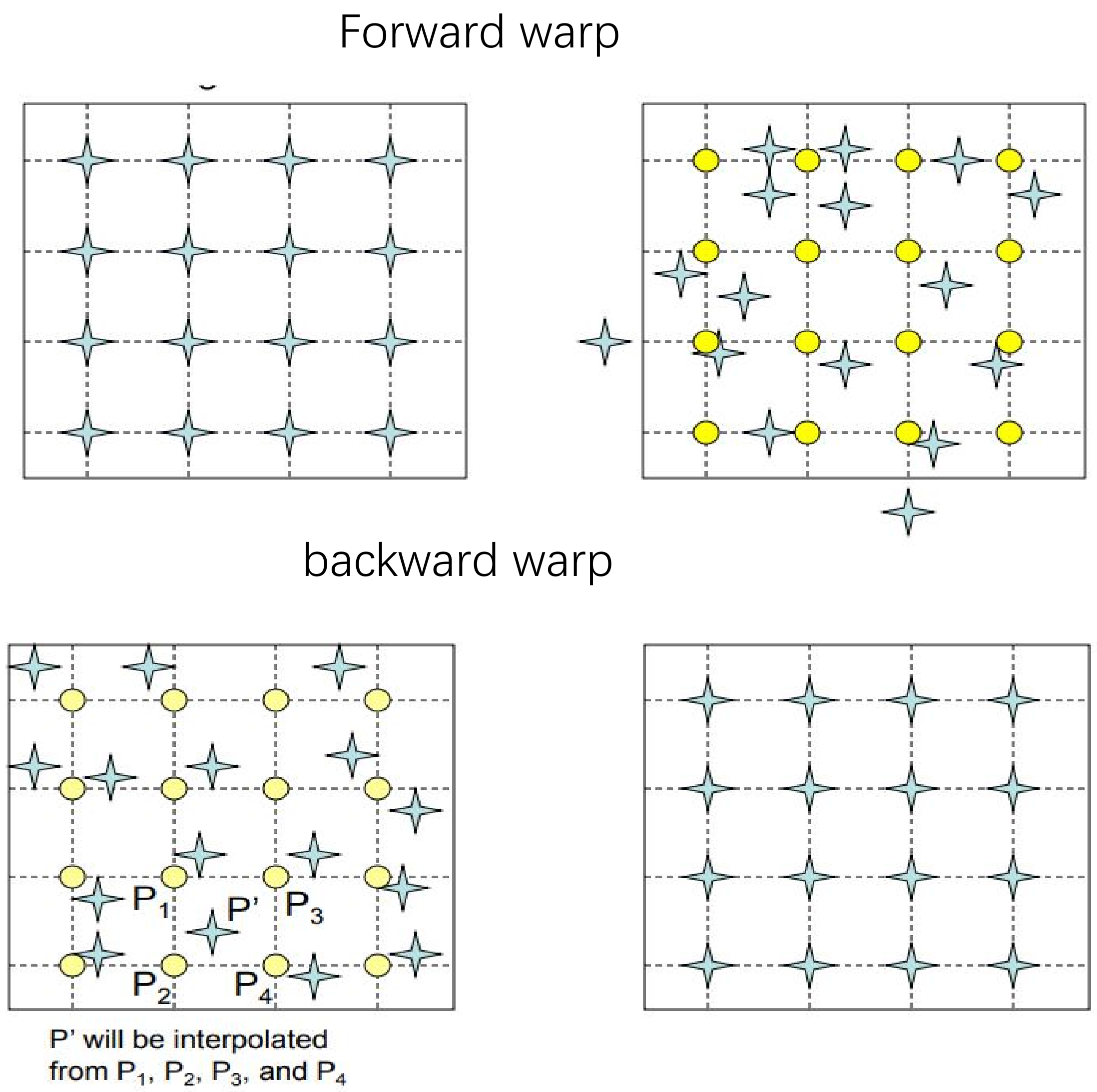
- 传统方法
  - $I(x,y,t) = I(x+dx,\ y+dy,\ t+\mathrm{dt})$
  - $I(x,y,t) = I(x,y,t) + \frac{\partial I}{\partial x}\mathrm{dx} + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \epsilon$
  - $\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t}\frac{dt}{dt} = 0$
  - $I_x u + I_y v + I_t = 0$
- 深度学习方法
  - FlowNet
  - PWC-Net

Forward warp vs Backward warp

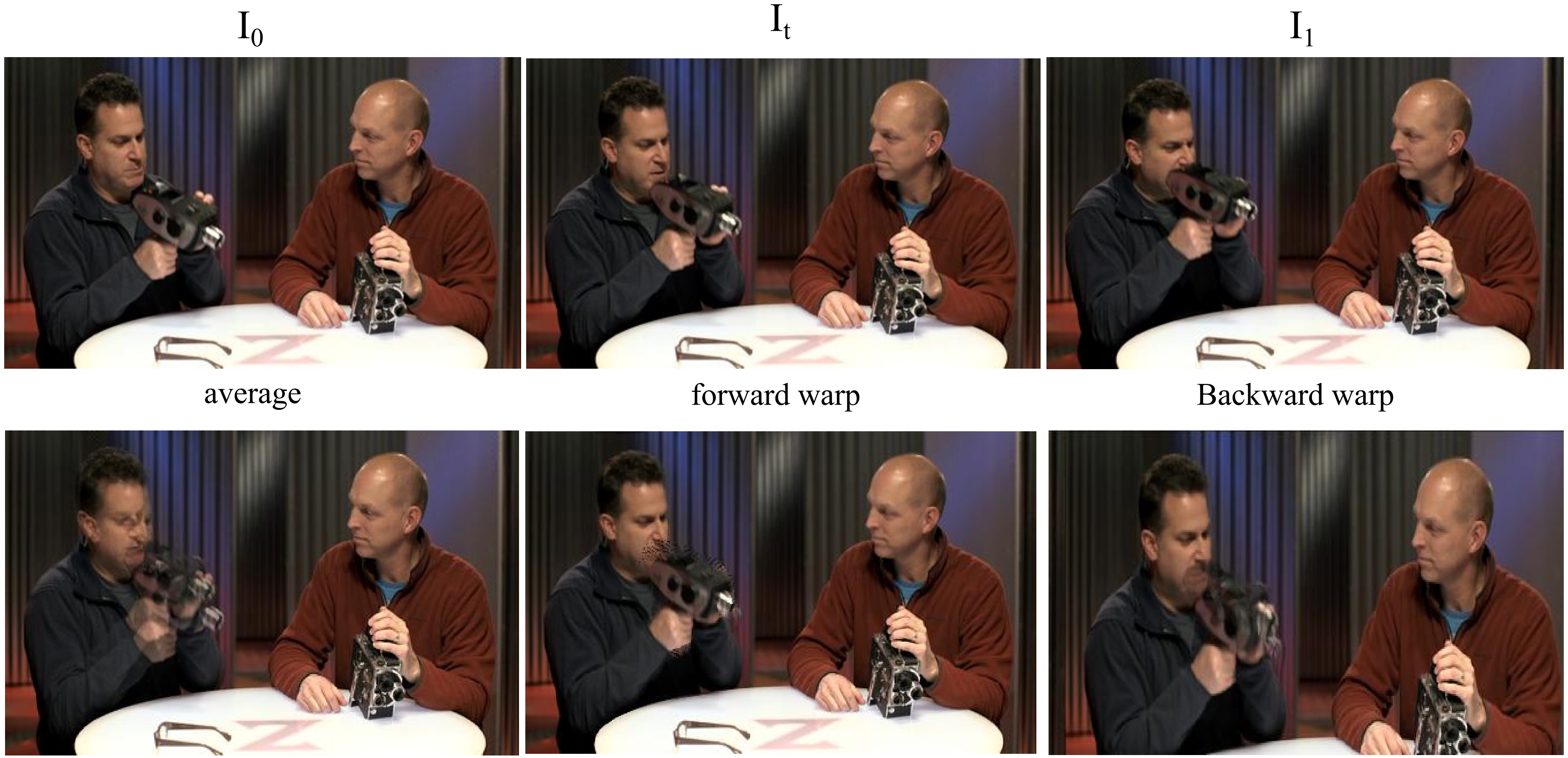
$s(x', y')$, $d(x, y)$, $f(u, v)$

$$x = x' + u$$
$$y = y' + v$$

$$x' = x - u$$
$$y' = y - v$$



方法对比



视频插帧研究现状

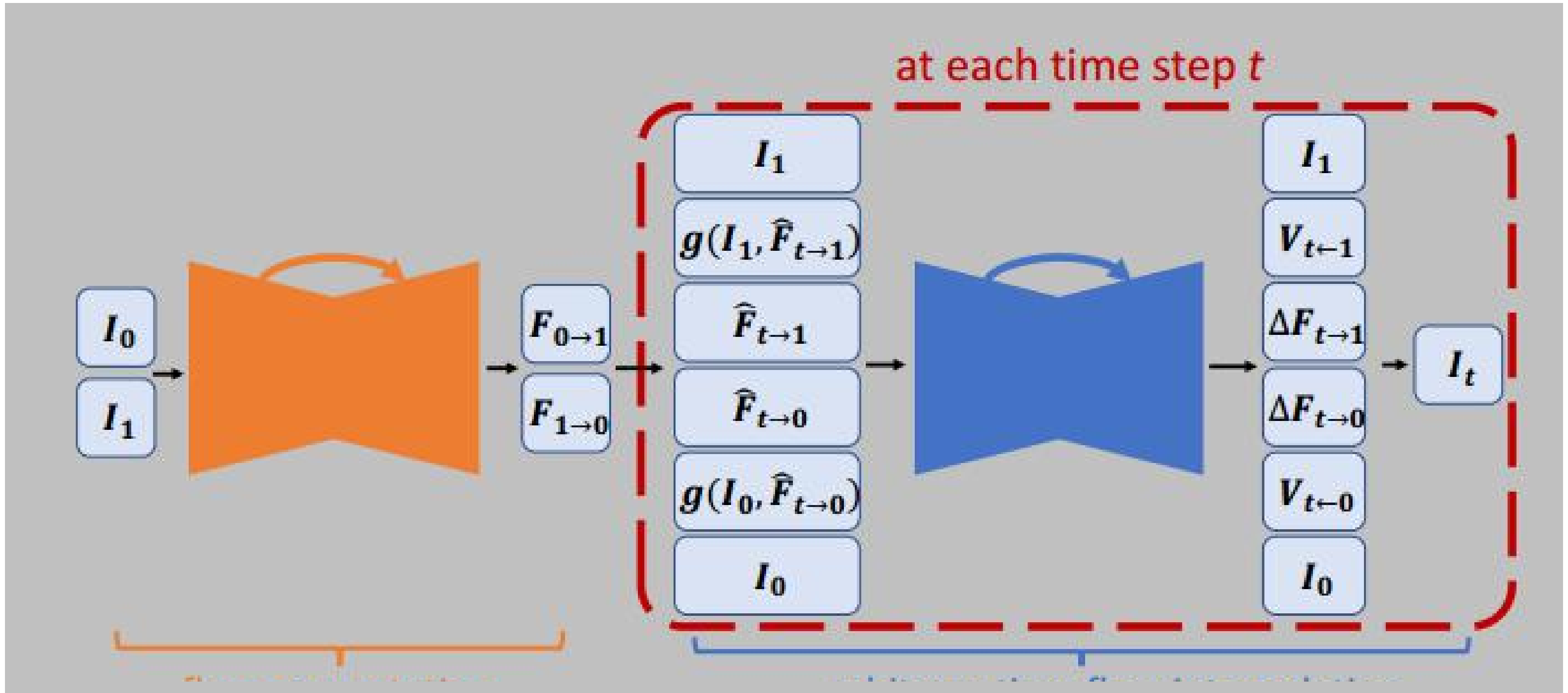
基于光流+backwarp

- super slomo
- RIFE

基于kernel+deformable convolution

- AdaCof

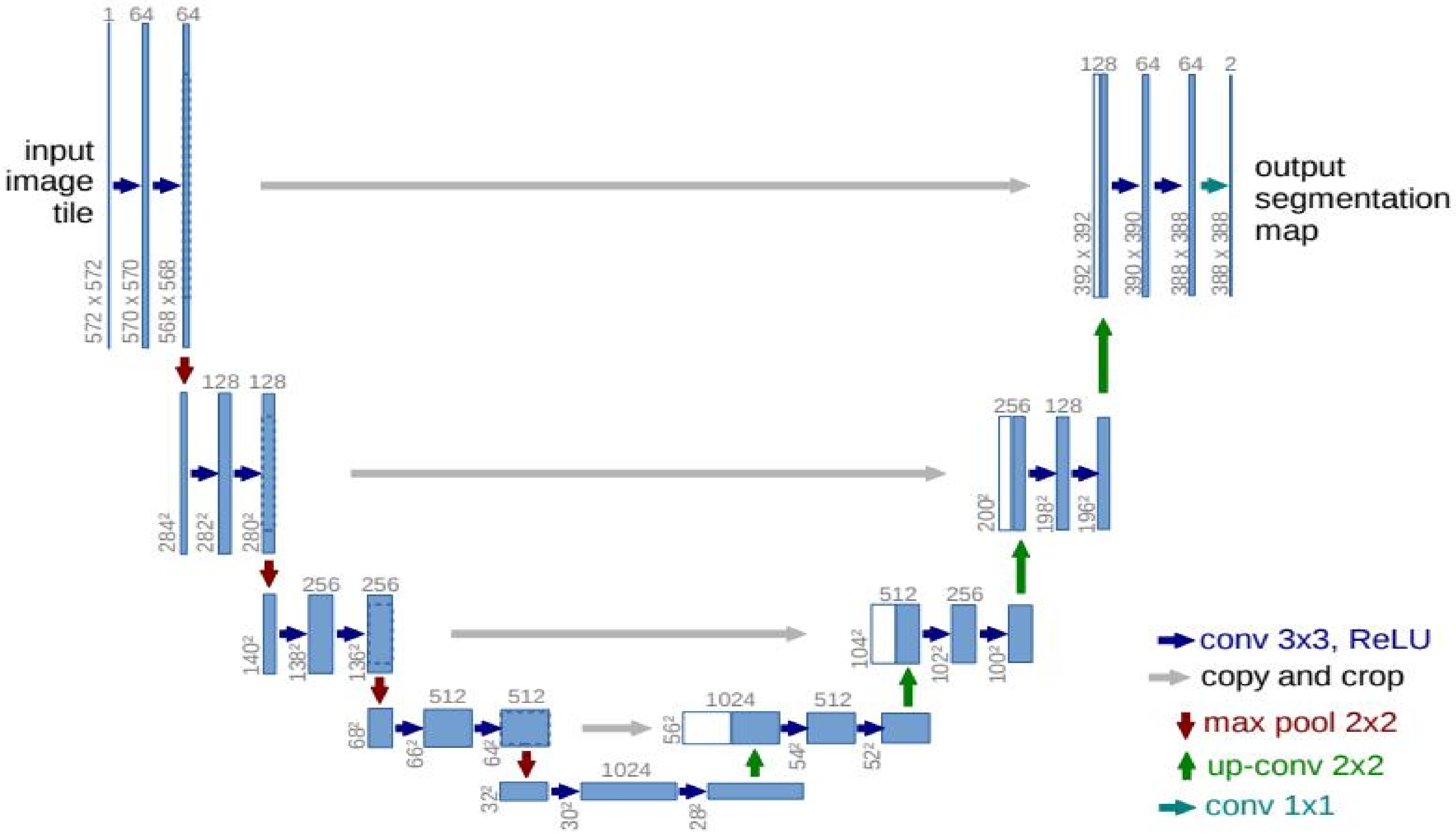
Super slomo: High quality estimation of multiple intermediate frames for video interpolation



$$\hat{I}_t = \frac{1}{Z} \odot \left((1-t) V_{t\leftarrow 0} \odot g(I_0, F_{t\to 0}) + t V_{t\leftarrow 1} \odot g(I_1, F_{t\to 1})\right)$$

Jiang H, Sun D, Jampani V, et al. Super slomo: High quality estimation of multiple intermediate frames for video interpolation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 9000-9008.

Unet



input image tile
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1
1 64 64
572 x 572
570 x 570
568 x 568
128 128
256 256
512 512
1024
1024 512
512 256
256 128
128 64 64 2
392 x 392
390 x 390
388 x 388
388 x 388

损失函数

$$l = \lambda_r l_r + \lambda_p l_p + \lambda_w l_w + \lambda_s l_s. \tag{7}$$

$$l_r = \frac{1}{N}\sum_{i=1}^{N} \|\hat{I}_{t_i} - I_{t_i}\|_1. \tag{8}$$

$$l_p = \frac{1}{N}\sum_{i=1}^{N} \|\phi(\hat{I}_t) - \phi(I_t)\|_2, \tag{9}$$

$$l_w = \|I_0 - g(I_1, F_{0\to 1})\|_1 + \|I_1 - g(I_0, F_{1\to 0})\|_1 + \tag{10}$$

$$\frac{1}{N}\sum_{i=1}^{N} \|I_{t_i} - g(I_0, \hat{F}_{t_i\to 0})\|_1 + \frac{1}{N}\sum_{i=1}^{N} \|I_{t_i} - g(I_1, \hat{F}_{t_i\to 1})\|_1.$$

$$l_s = \|\nabla F_{0\to 1}\|_1 + \|\nabla F_{1\to 0}\|_1.$$

实验结果

Table 4: Results on the *UCF101* dataset.

	PSNR	SSIM	IE
Phase-Based [18]	32.35	0.924	8.84
FlowNet2 [1, 9]	32.30	0.930	8.40
DVF [15]	32.46	0.930	8.27
SepConv [20]	33.02	0.935	8.03
Ours (Adobe240-fps)	32.84	0.935	8.04
Ours	**33.14**	**0.938**	**7.80**

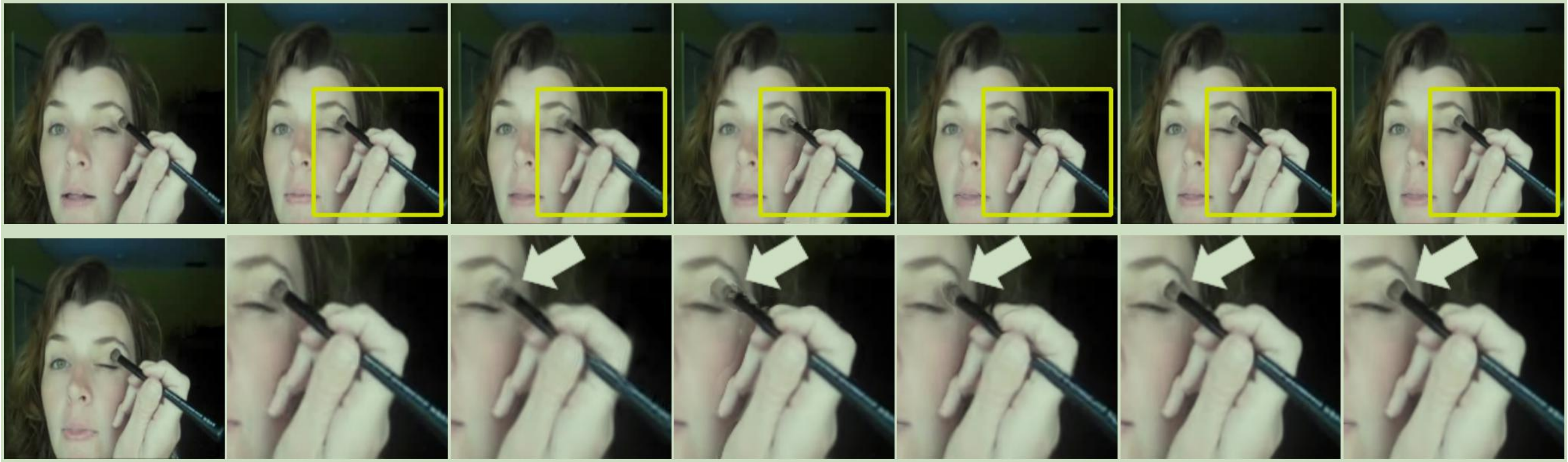
Table 5: Results on the *slowflow* dataset.

	PSNR	SSIM	IE
Phase-Based [18]	31.05	0.858	8.21
FlowNet2 [1, 9]	34.06	**0.924**	**5.35**
SepConv [20]	32.69	0.893	6.79
Ours	**34.19**	**0.924**	6.14

Table 6: Results on the high-frame-rate *Sintel* dataset.

	PSNR	SSIM	IE
Phase-Based [18]	28.67	0.840	10.24
FlowNet2 [1, 9]	30.79	0.922	5.78
SepConv [20]	31.51	0.911	6.61
Ours	**32.38**	**0.927**	**5.42**

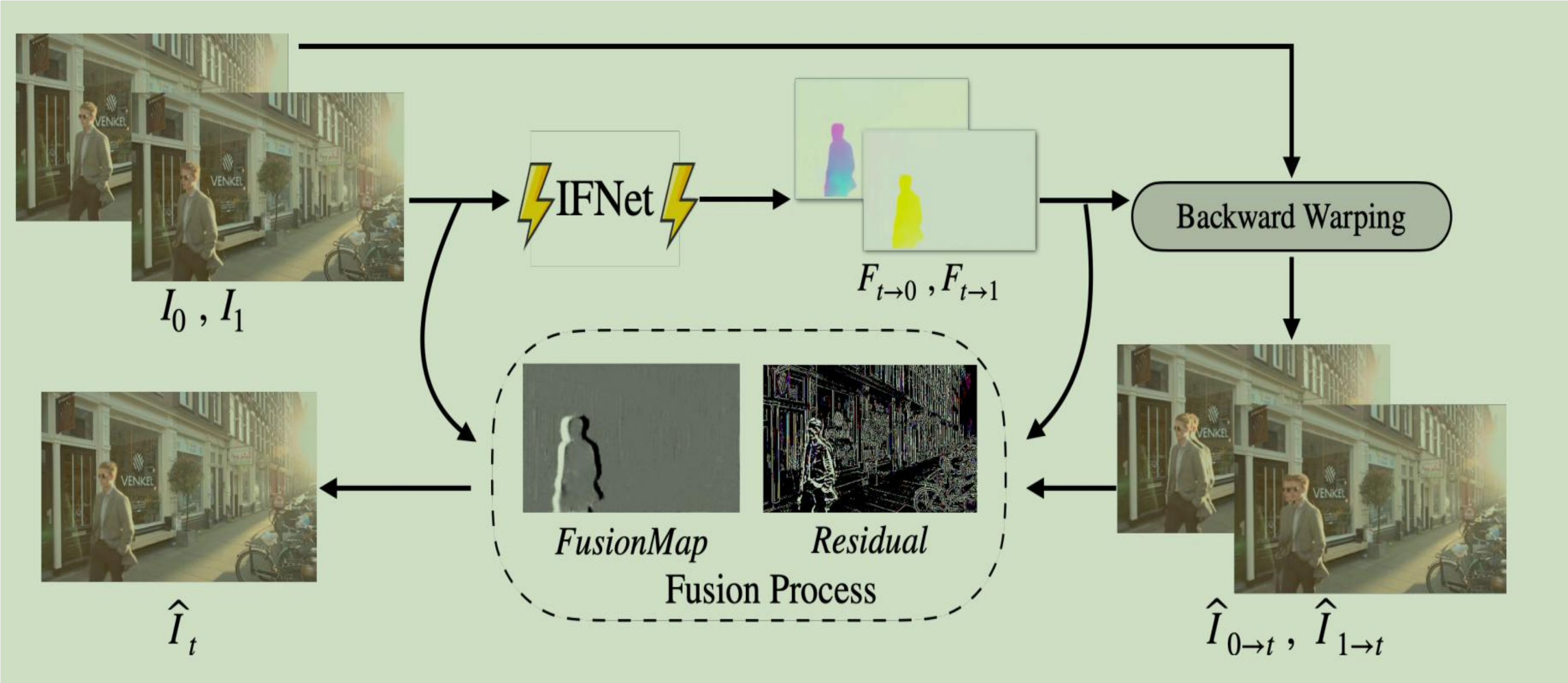
主观结果



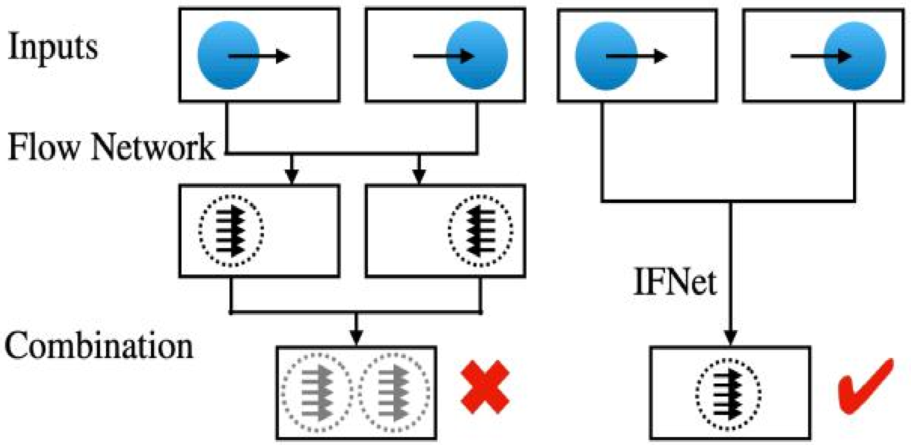
2 input images	actual in-between	PhaseBased [18]	FlowNet2 [1, 9]	DVF [15]	SepConv [20]	ours

RIFE: Real-Time Intermediate Flow Estimation for Video Frame Interpolation

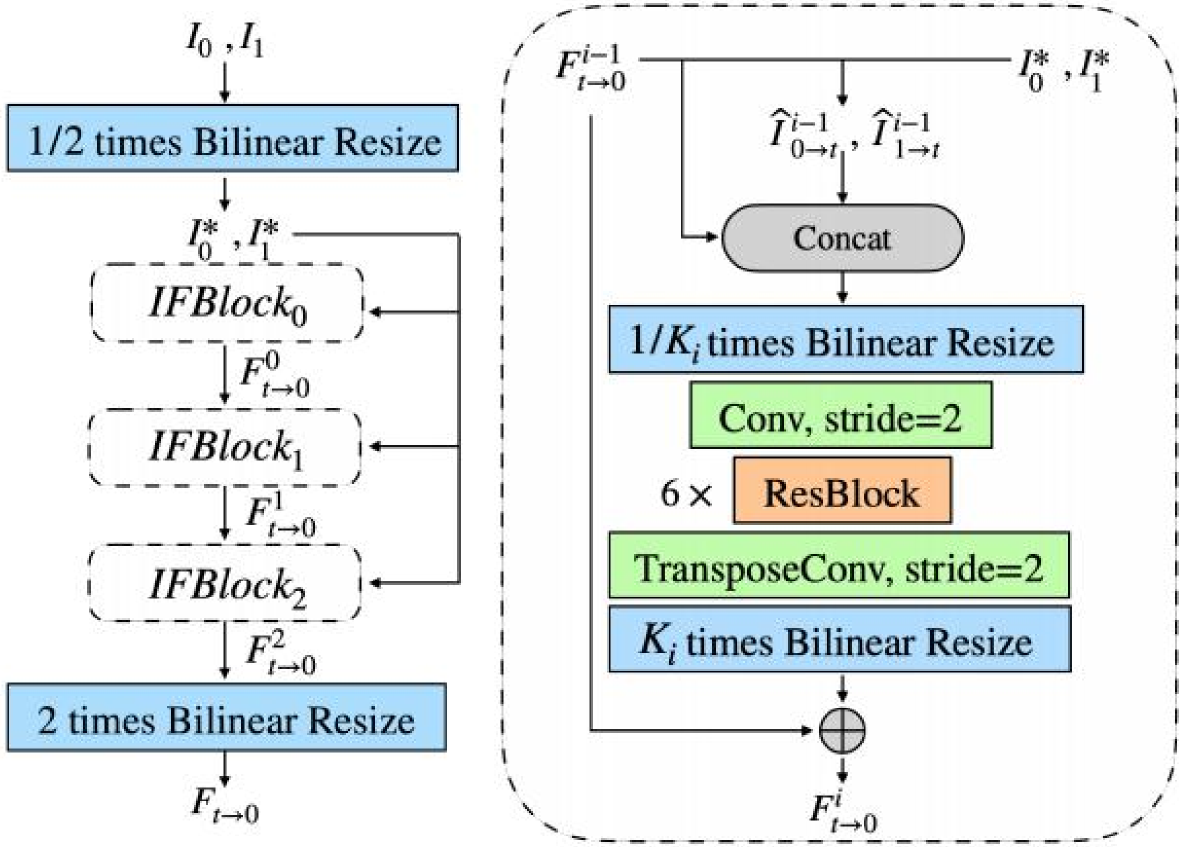
- 多尺度光流
- 多尺度特征融合
- 残差信息融合



IFNet光流估计



Inputs
Flow Network
Combination
IFNet



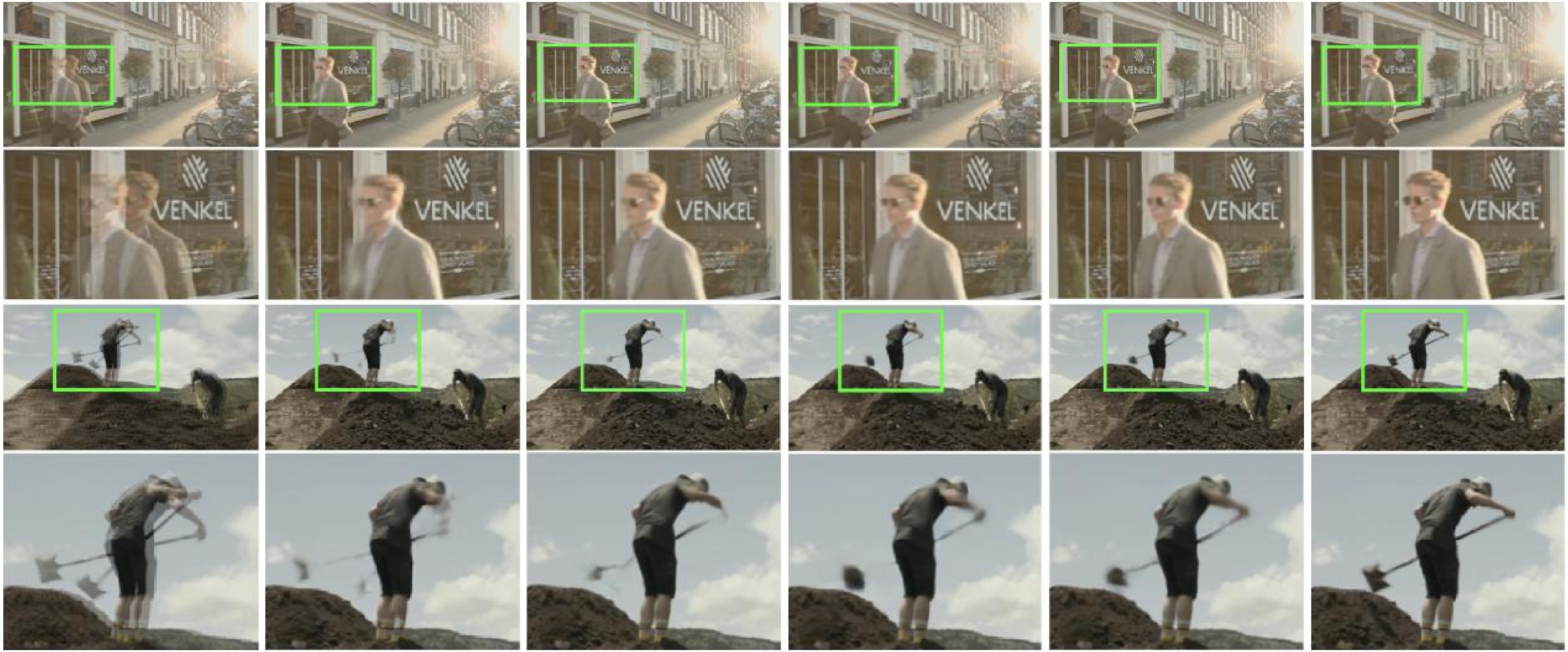
I_0, I_1
1/2 times Bilinear Resize
I_0^*, I_1^*
$IFBlock_0$
$F_{t\to0}^0$
$IFBlock_1$
$F_{t\to0}^1$
$IFBlock_2$
$F_{t\to0}^2$
2 times Bilinear Resize
$F_{t\to0}$
$F_{t\to0}^{i-1}$
I_0^*, I_1^*
$\hat{I}_{0\to t}^{i-1}, \hat{I}_{1\to t}^{i-1}$
Concat
$1/K_i$ times Bilinear Resize
Conv, stride=2
6× ResBlock
TransposeConv, stride=2
K_i times Bilinear Resize
$F_{t\to0}^i$

实验结果

Method	# Parameters (Million)	Runtime (ms)	UCF101 [28]		Vimeo90K [35]		Middlebury [1]	HD [3]
			PSNR	SSIM	PSNR	SSIM	IE	PSNR
TOFlow [35]	**1.1**	430	34.58	0.967	33.73	0.968	2.15	29.37
SepConv-$\mathcal{L}_1$[24]	21.6	200	34.78	0.967	33.79	0.970	2.27	30.87
MEMC-Net [3]	70.3	121	35.01	0.968	34.40	0.970	2.12	31.60
DAIN [2]	24.0	125	35.00	0.968	34.71	0.976	2.04	31.64
CAIN [8]	42.8	32*	34.91	0.969	34.65	0.973	2.28	30.70*
SoftSplat [23]	7.7	135	**35.39**	**0.970**	36.10	0.980	-	-
BMBC [26]	11.0	770	35.15	0.969	35.01	0.976	-	-
RIFE (Ours)	10.4	**21**	35.14	0.969	35.69	0.978	2.05	32.04
RIFE-Large (Ours)	22.9	90	35.33	**0.970**	**36.24**	**0.981**	**1.98**	**32.18**

*: use officially released models to produce results

实验结果



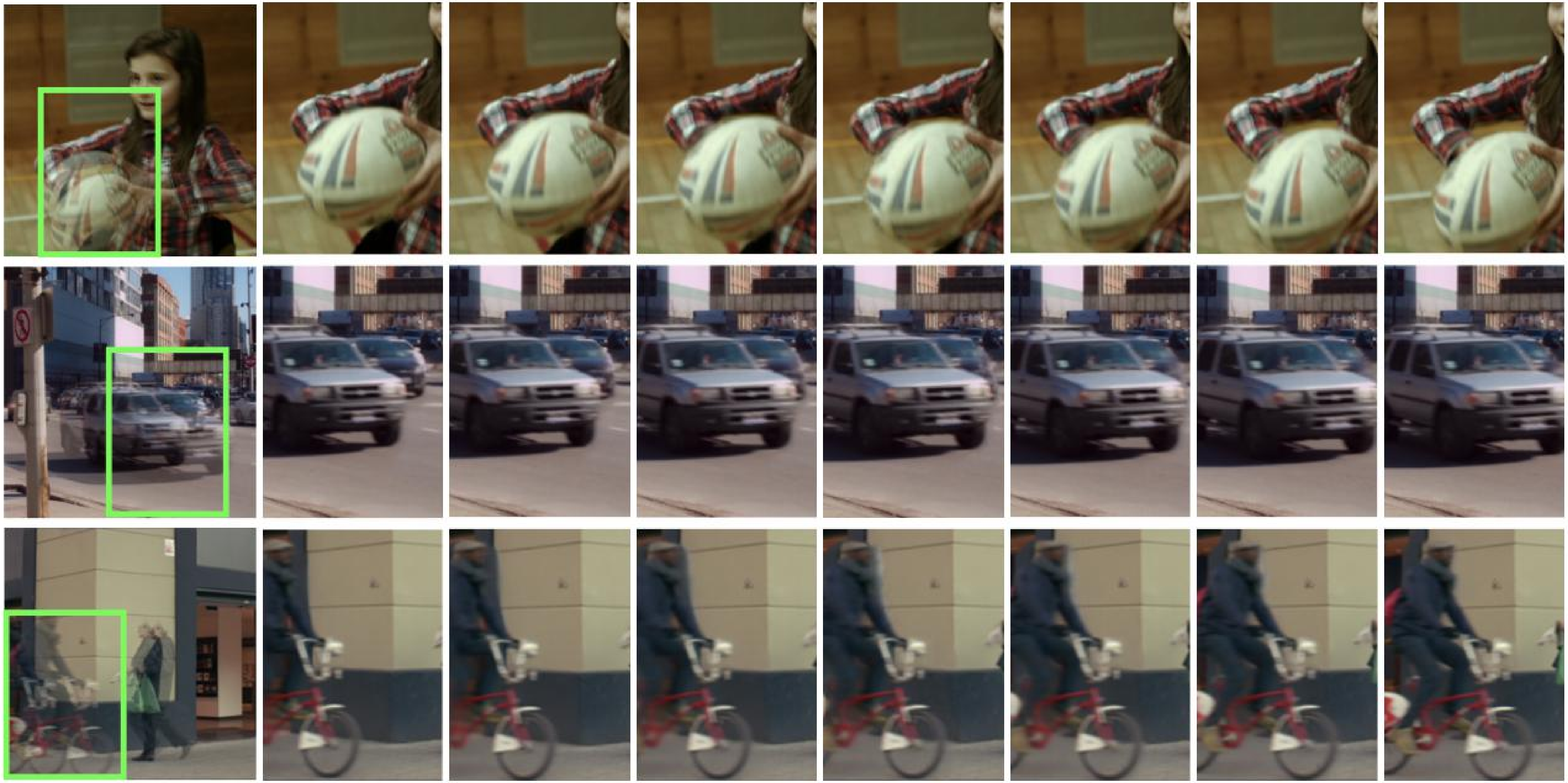
Inputs (Overlay) | SepConv-L_1 [25] | DAIN [2] | CAIN [8] | RIFE (Ours) | GT

复杂度与收益

Scale Setting	RIFE	*1.5C*	*2F*	RIFE-Large
UCF101 PSNR	35.14	35.26	35.32	**35.33**
Vimeo90K PSNR	35.69	35.88	36.08	**36.24**
Middlebury IE	2.03	2.03	1.99	**1.98**
HD PSNR	32.04	32.13	31.96	**32.18**
# Parameters*	**10.4M**	22.9M	**10.4M**	22.9M
Runtime*	**36ms**	65ms	126ms	196ms
Complexity*	**83G**	185G	322G	724G

*: measure the whole algorithm on 720p videos

多帧插值



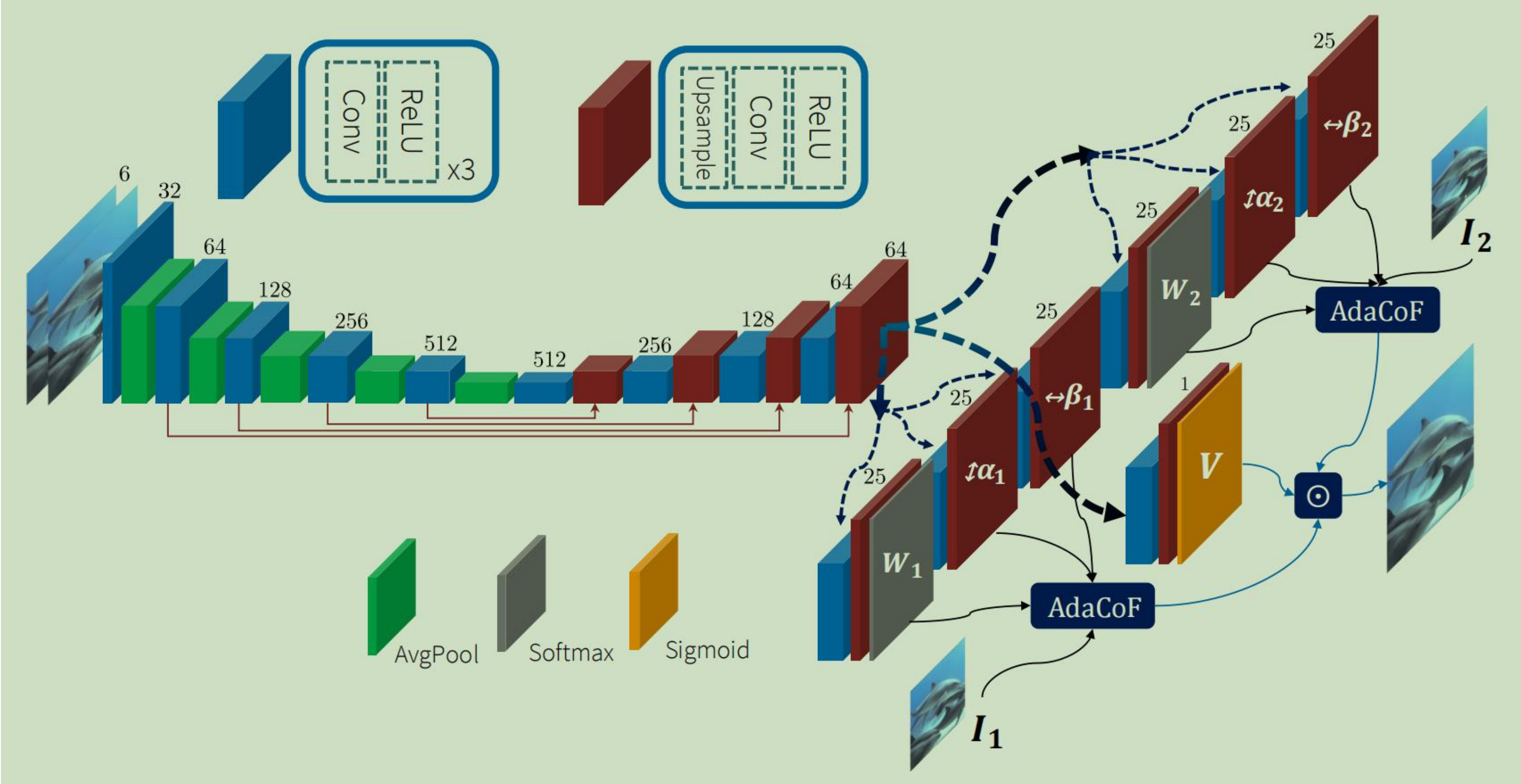
Inputs (Overlay) $\widehat{I}_{0.125}$ $\widehat{I}_{0.25}$ $\widehat{I}_{0.375}$ $\widehat{I}_{0.5}$ $\widehat{I}_{0.625}$ $\widehat{I}_{0.75}$ $\widehat{I}_{0.875}$

AdaCoF: Adaptive Collaboration of Flows for Video Frame Interpolation

Deformable-convolution

权重学习

$$\hat{I}(i,j) = \sum_{k=0}^{F-1}\sum_{l=0}^{F-1} W_{k,l} I(i+k+\alpha_{k,l}, j+l+\beta_{k,l})$$



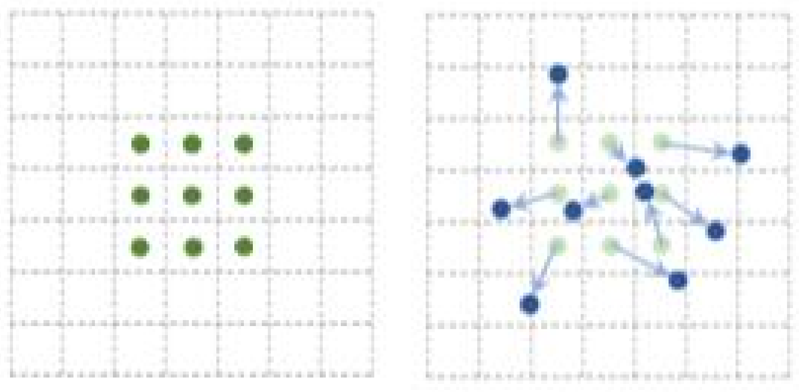
Deformable Convolution

Convolution

$$\hat{I}(i,j) = \sum_{k=0}^{F-1}\sum_{l=0}^{F-1} W_{k,l} I(i+k, j+l)$$

Deformable Convolution

$$\hat{I}(i,j) = \sum_{k=0}^{F-1}\sum_{l=0}^{F-1} W_{k,l} I(i+k+\alpha_{k,l}, j+l+\beta_{k,l})$$



(a) (b)

实验结果

	Middlebury		UCF101		DAVIS	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$F = 1$	32.879	0.956	33.449	0.967	24.787	0.828
$F = 3$	35.212	0.975	34.728	0.973	26.535	0.867
$F = 5$	35.715	0.978	**35.063**	**0.974**	26.636	0.868
$F = 7$	35.927	0.979	34.974	**0.974**	**26.987**	**0.873**
$F = 9$	**36.019**	**0.980**	35.012	0.973	**27.029**	**0.875**
$F = 11$	**36.094**	**0.981**	**35.024**	**0.974**	26.941	**0.873**

Table 2: Experimental result on kernel size F.

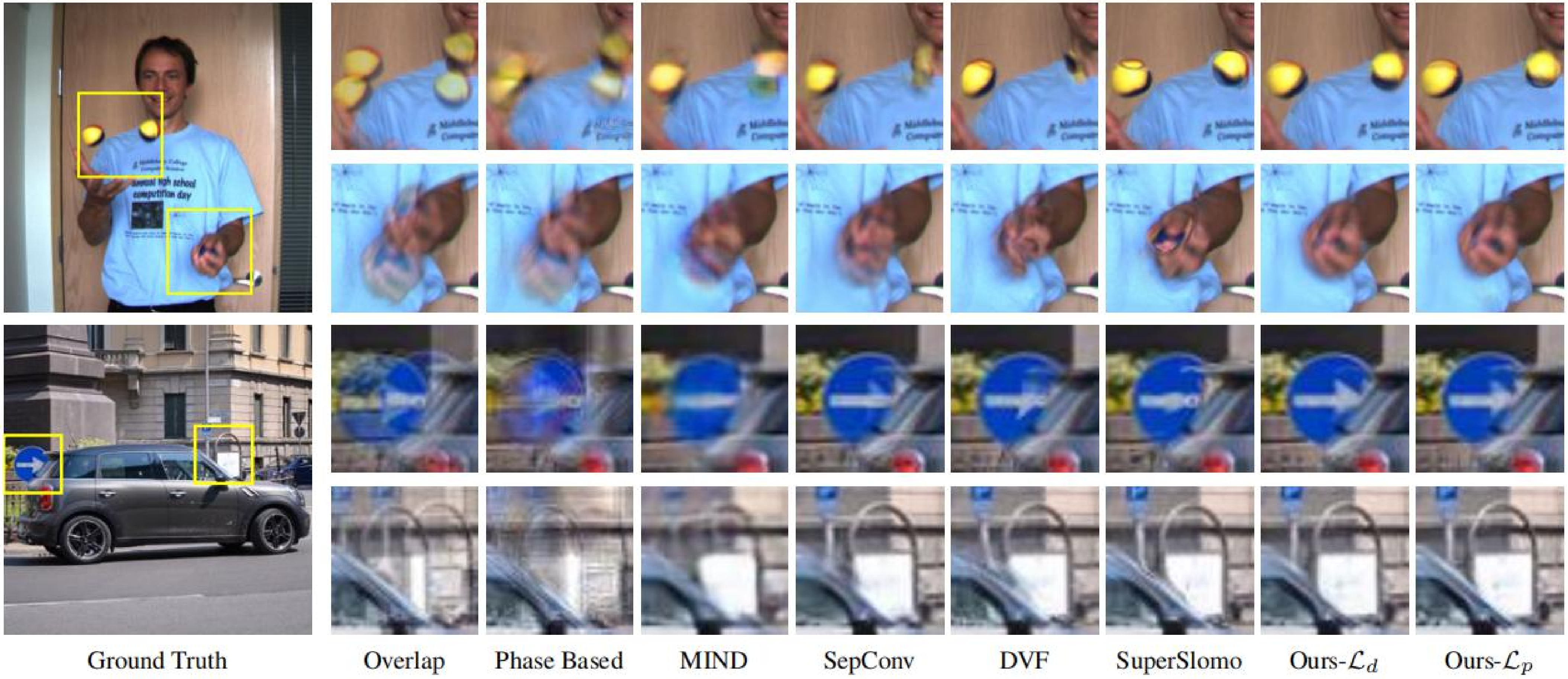
	Middlebury		UCF101		DAVIS	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$d = 0$	35.489	0.977	35.032	**0.974**	**26.710**	**0.870**
$d = 1$	**35.715**	**0.978**	**35.063**	**0.974**	26.636	0.868
$d = 2$	**35.876**	**0.980**	**35.099**	**0.974**	**26.910**	**0.870**

Table 3: Experimental result on dilation d.

实验结果

	Middlebury		UCF101		DAVIS	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Overlapping	27.968	0.879	30.445	0.935	21.922	0.740
Phase Based [32]	31.117	0.933	32.454	0.953	23.465	0.800
MIND [28]	31.346	0.943	32.437	0.963	25.570	0.852
SepConv [35]	35.521	0.977	34.735	0.973	26.258	0.861
DVF [27]	34.340	0.971	34.465	0.972	25.880	0.858
SuperSlomo [20]	34.234	0.972	34.055	0.970	25.699	0.858
Ours	**35.715**	**0.978**	**35.063**	**0.974**	**26.636**	**0.868**
Ours +	**36.139**	**0.981**	**35.048**	**0.974**	**27.070**	**0.874**

主观效果



Ground Truth
Overlap
Phase Based
MIND
SepConv
DVF
SuperSlomo
Ours-$\mathcal{L}_d$
Ours-$\mathcal{L}_p$

视频插帧存在的问题

■ 插帧效果

- 生成帧清晰度下降
- 低帧率插帧出现模糊、重影、形变

模糊

重影

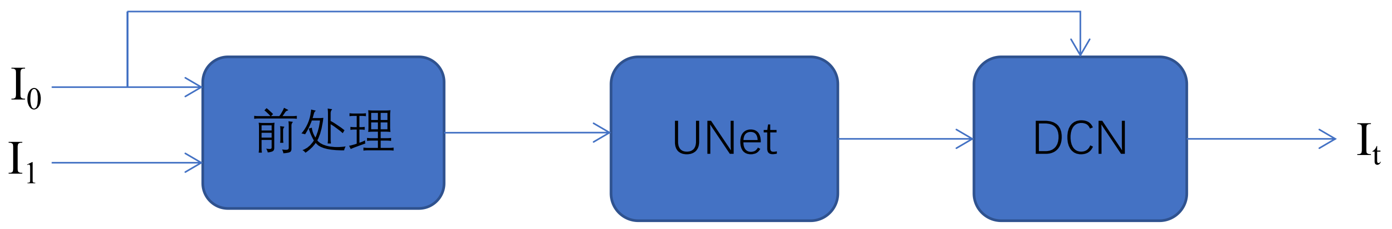
形变

■ 可用性受限

模型	大小	GFLOPs	推理速度 720p nvidia 1060
SuperSlomo	151MB	14G	200ms
RIFE	114MB	724G	190ms

视频插帧方案

- 前处理
- 单方向光流
- Kernel+Deformable convolution



前处理

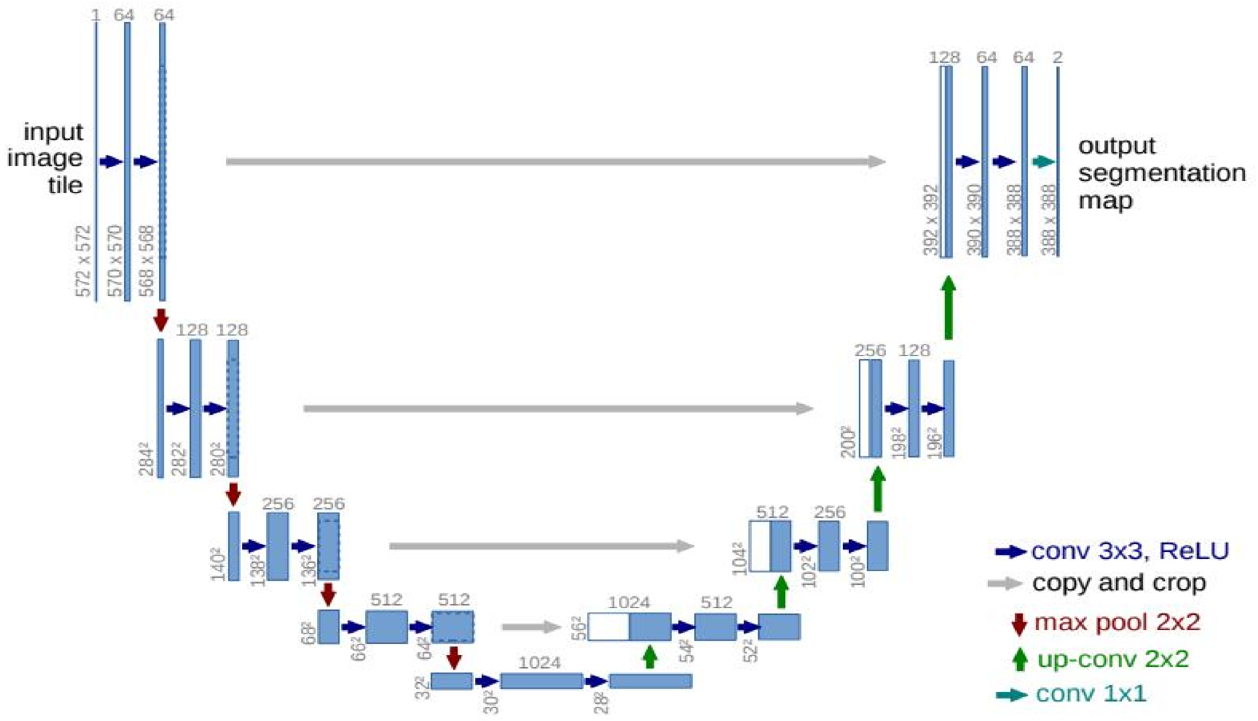
规避大幅度运动
防止效果回退

I_0 I_t I_1

Unet

多尺度
运算量集中头尾部
参数里集中中部

计算单向光流
add代替cat
常规conv2d
插值法代替decon



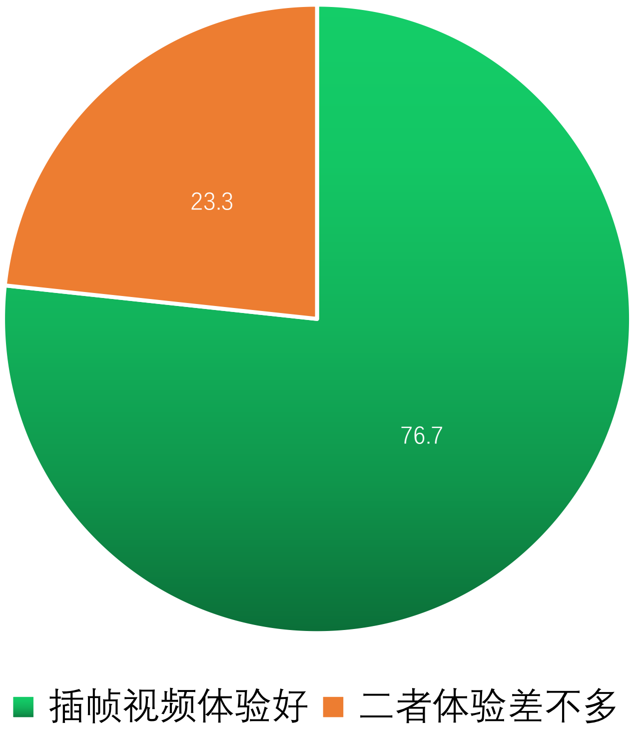
Demo

主观评估

12人进行主观评估

30个视频片段，帧率6fps~15fps

A-B对比



总结

- 更低帧率视频的插帧
- 多帧插帧
- 处理高分辨率