# Depth-Aware Video Frame Interpolation(译)

Relu/邱浩轩

# 深度感知视频帧插值

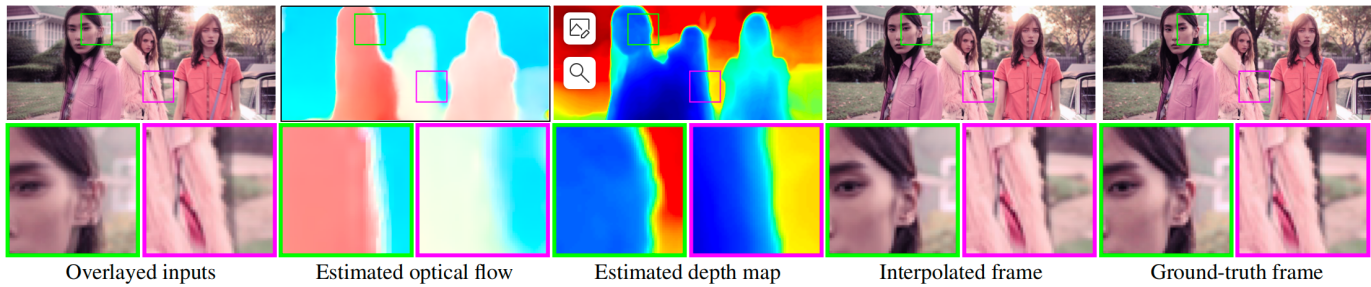


图 1

## 贡献

1. 我们发现深度感知映射层(depthaware flow projection layer)能够更好地合成更近的目标。
2. 我们提出了深度感知视频插值的方法(depth-aware video frame interpolation)，其整合了光流、本地插值内核、深度映射和可学习的分层特征以进行高质量的插值合成。
3. 我们证明，与最先进的方法相比，我们提出的模型更有效、更高效、更紧凑。

## 相关工作

视频帧插值是一个长期存在的话题，在文献中得到了广泛的研究，在这个部分，我们重点讨论最近的基于深度学习的算法，此外我们讨论了深度估计的相关主题。

### 视频帧插值

作为基于CNN的方法的先驱 Long et al.训练通用CNN直接合成中间帧。然而，他们的结果严重模糊，因为通用CNN无法捕捉自然图像和视频的多模态分布。然后Liu et al.提出深体素流(deep voxel flow)，一种跨越空间和时间的三维光流(3D optical flow)，以基于三线性采样来warp输入帧。虽然从流中合成的帧的模糊性较小，但是对于大运动的场景来说，流估计仍然具有挑战性。不准确的流动可能会导致严重的失真和视觉伪影。

除了依赖光流，AdaConv和SepConv方法估计空间自适应插值核，以合成来自大邻域的像素。然而这些kernel-based的方法通常需要高内存占用，并需要很大的计算负载。不久前， Bao et al.将flow-based和kernel-based的方法集成到端到端网络中，以继承双方的优点

输入帧首先被光流warped，然后通过自适应warping层中学习的插值kernels进行采样。

现有方法通过估计遮挡掩码、提取上下文特征、学习大型局部插值kernels隐式处理遮挡。

相比之下，我们通过利用流投影层中的深度信息来明确检测遮挡。此外，我们将深度图与学习的层次特征(hierarchical features)相结合，作为上下文信息，以合成输出帧。

### 深度估计

深度是理解场景3D几何的关键视觉信息之一，并在几个识别任务中被利用，比如语义分割和目标检测。传统的方法需要立体的图像作为输入去估计差异。最近，一些learning-based的方法旨在从单个图像中估计深度。在这项工作中，我们使用Chen et al.的模型，那是一个在MegaDepth数据集上训练的hourglass网络，用于从输入帧预测深度图。我们表明，深度网络的初始化对于推断遮挡至关重要。然后，我们与其他子模块共同微调depth网络，以进行帧插值。因此，我们的模型学习了warping和插值的relative depth。

我们注意到，几种方法通过利用跨任务约束和一致性来共同估计光流和深度。虽然我们提出的模型还同时估计了光流和深度，但我们的流和深度针对帧插值进行了优化，这可能与像素运动和场景深度的真实值不同。

## 深度感知视频帧插值

在本节中，我们首先概述了我们的帧插值算法。然后，我们介绍我们提出的深度感知流投影层(depth-aware flow projection layer)，是处理流聚合遮挡的关键组件。最后，我们描述了所有子模块的设计，并提供了我们提出的模型的实现细节。

### 算法概述

给定两个输入帧和，x∈[1, H] x [1, W] 表示2D的图像平面坐标，H，W分别是图像的高和宽，我们的目标是在时间t∈[0, 1]内合成一个中间帧。我们提出的方法要求光流warp输入帧，以合成中间帧。我们首先估计双向光流( the bi-directional optical

flows)，分别表示为和，为了合成中间帧，这里有两种常见的策略：

首先，第一种可以应用前向warping( the forward warping)去warp基于的和基于的。但是，the forward warping会导致被warp的图像上产生空洞。

第二个策略是第二种策略是近似中间流，到然后用反向warp(the

backward warping)去采样输入的帧。为了近似中间流，可以从和中相同的网格坐标中借用流向量，或者聚合通过相同位置的流向量。

在这个工作中，我们采纳了Bao et al的流映射层( the flow projection layer)，以便于聚合流动矢量，同时考虑深度顺序来检测遮挡。

在获取中间流之后，我们用基于光流和插值kernels(interpolation kernels)的自适应warp层(adaptive warping layer)将输入帧、上下文特征和深度图像进行warp操作。

### 深度感知流映射( Depth-Aware Flow Projection)

这个flow projection layer通过“反转”在时间t通过x的流向量来近似给定位置x的中间流。如果流在时间t通过x，可以通过来近似，同样的，我们可以通过来近似。

然而如图2中的1维时空所示，多个流向量可以在时间t投影到同一位置。除了通过简单平均聚合这些流，我们建议考虑聚合的深度排序。特别的，我们假定是的深度图，表示在时间t处穿过位置x的像素集。这个被映射的流定义为

其中权重是深度的倒数：

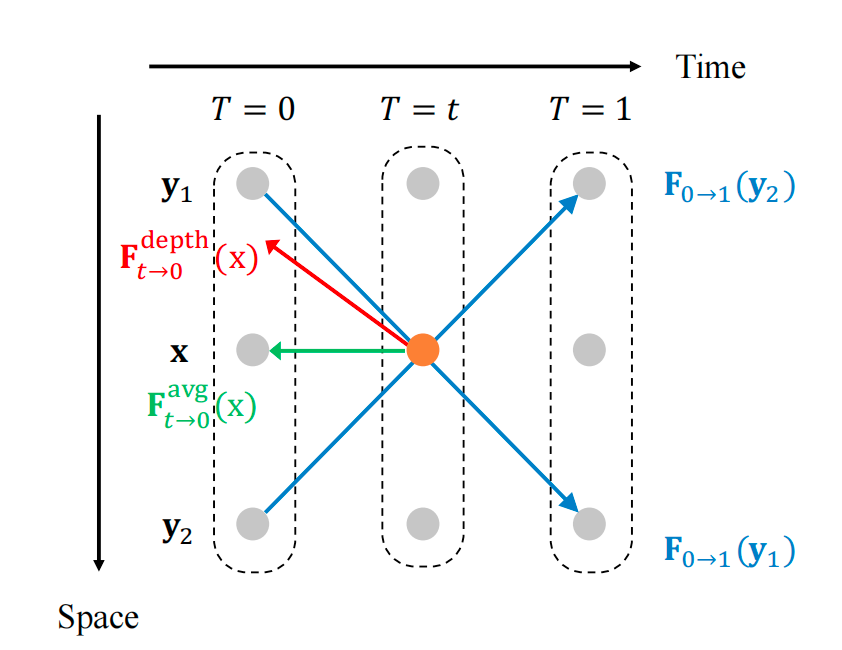


图 2深度感知流映射( Depth-Aware Flow Projection)现有的流投影方法获得一个平均流向量，可能不指向正确的对象或像素。相反，我们根据深度值重新编写流，并生成指向更近像素的流向量。

同样，映射流可以从流和深度图获得。通过这种方式，这些映射流更倾向于采样更近的对象并减少具有较大深度值的被遮挡像素所带来的贡献(影响)。如图2所示，这个流映射被用于生成平均的流向量(绿色的箭头)，这个流向量可能无法指向正确的像素进行采样。相反，这个来自我们深度感知流映射层(Depth-Aware Flow Projection)的映射流(红色箭头)指向具有较小深度值的像素点。

另一方面，可能存在一些位置，其中没有任何流向量通过，导致中间流中出现空洞。为了填补这些空洞，我们使用从外到内的策略：空洞位置的流是通过平均来自其邻域可用流来计算的：

这个是x的4领域，通过公式(1)，(3)，我们获得了用于warp输入帧的密集中流场(dense intermediate flow fields)和。

提出的深度感知流映射层是完全可微分的，因此在训练过程中可以同时优化光流和深度估计网络。我们在补充材料中提供了深度感知流映射的反向传播细节。

### 视频帧插值(Video Frame Interpolation)

我们提出的模型由以下子模块组成：流估计，深度估计，上下文提取，kernel估计以及帧合成网络。我们用深度感知流映射层(depth-aware flow projection layer)去获取中间流然后warp输入帧、深度图以及自适应warp层(adaptive warping layer)中的上下文特征。最后，帧合成网络通过残差学习生成输出帧。我们在图3展示了整体的网络架构。以下我们描述每个子网络的详细信息。

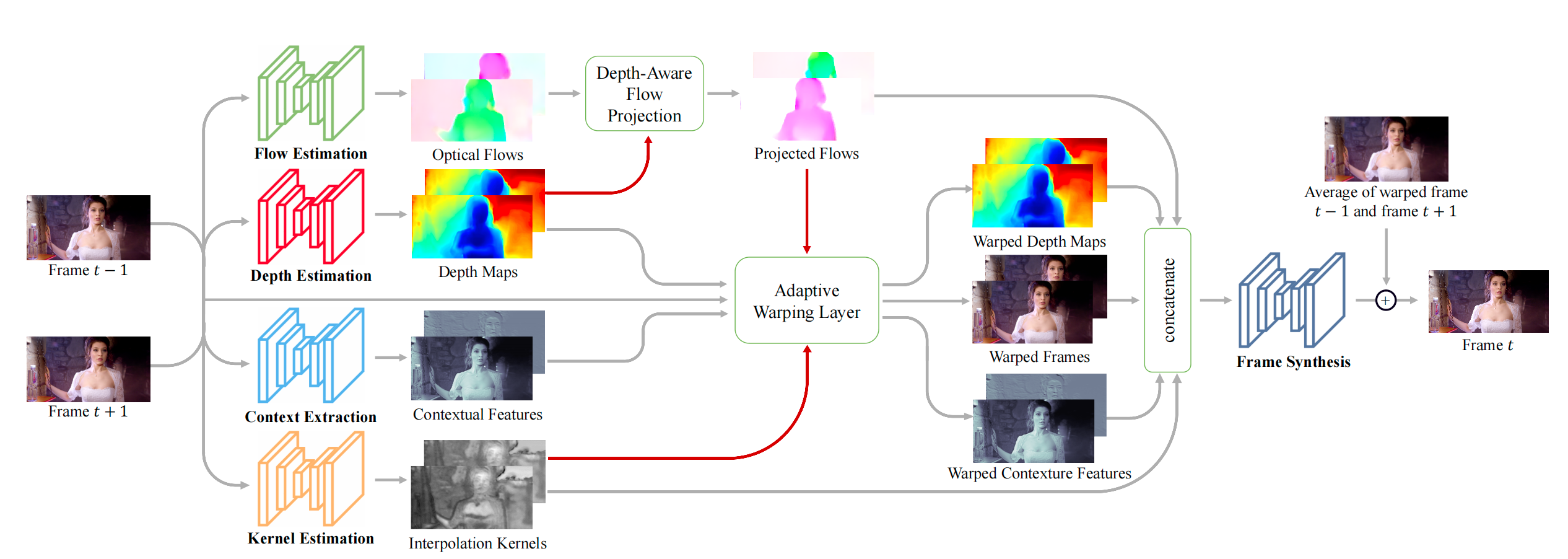


图 3 提出的深度感知视频帧插值模型(depth-aware video frame interpolation model)的整体架构。给定两个输入帧，我们先估计它们的光流和深度图，并用深度感知流映射层去生成中间流。然后，我们采用自适应warp层根据流和空间可变插值kernels对输入帧、深度图和上下文特征进行warp。最后我们用一个帧合成网络去生成输出帧

#### 流估计(Flow estimation)

我们采用最先进的流模型PWC-Net作为我们的流估计网络(r flow estimation network)。由于在没有ground-truth监督的情况下学习光流是非常困难的，因此我们从预训练的PWC-Net中初始化我们的光流估计网络。

#### 深度估计(Depth estimation)

我们使用hourglass架构作为我们的深度估计网络。为了获得流映射的有意义的深度信息，我们从Li et al.的预训练模型中初始化深度估计网络。

#### 上下文提取(Context extraction)

在论文*W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang. MEMC-Net: Motion Estimation and Motion Compensation Driven Neural Network for Video Interpolation and Enhancement. arXiv, 2018*和论文*S. Niklaus and F. Liu. Context-aware synthesis for video frame interpolation. In CVPR, 2018* 中指出，用一个预训练的ResNet提取上下文信息，即第一个卷积层的特征图。然而，来自ResNet的特征是用于图像分类任务的，可能对视频帧插值不起作用。因此，我们建议学习上下文特征。具体来说，我们构建了一个上下文提取网络，其中包括一个7×7的卷积层和两个残差块，如图4所示。这个残差块由两个3🤔x3的卷基层和两个ReLU激活层构成。我们不使用任何normalization层，比如batch normalization。然后我们将第一个卷积层和两个残差块的特征连接起来，得到一个分层特征(hierarchical feature)。我们的上下文提取网络是从零开始训练的，因此可以学习到有效的视频帧插值上下文特征。

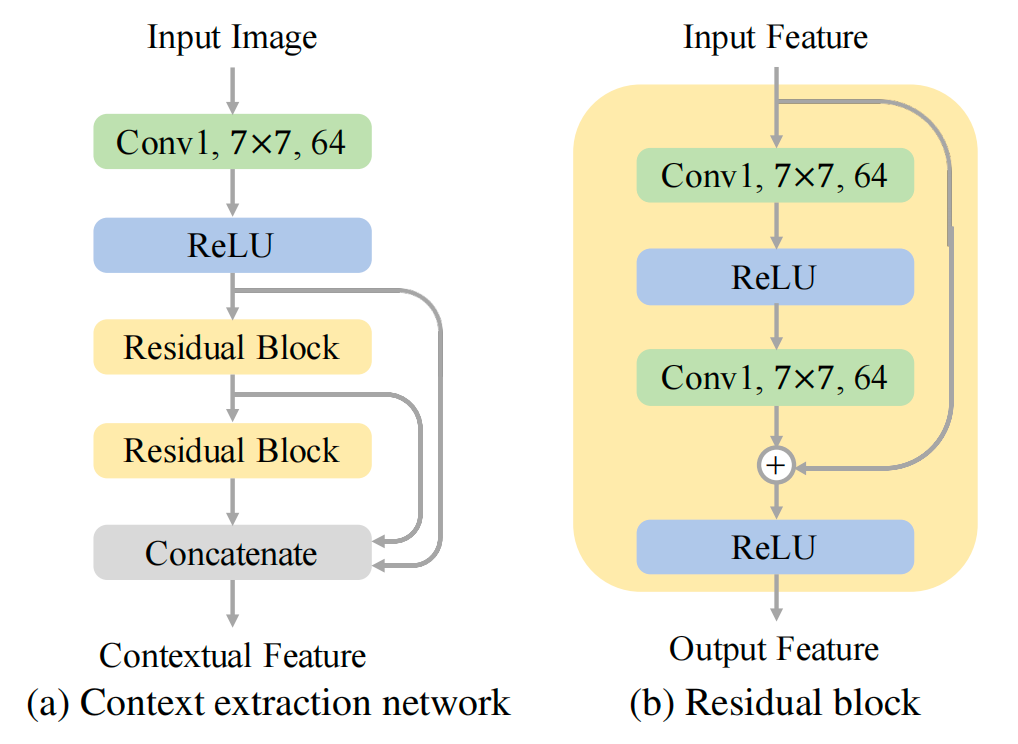


图 4 **上下文提取网络的结构。**我们不使用预训练分类网络的权重，而是从头开始训练我们的上下文提取网络，并学习用于视频帧插值的分层特征。

#### Kernel估计和自适应warp层

局部插值Kernel已被证明对于从大范围的局部邻域合成像素是有效的。Bao et al.进一步将插值Kernel和光流与自适应warp层进行整合。自适应warp层通过在局部窗口内对输入图像进行采样来合成新的像素，其中窗口的中心由光流指定。在这里，我们使用U-Net架构[30]来估计每个像素的4×4局部Kernel。通过深度感知流投影层生成的插值核和中间流，我们采用自适应变形层来对输入帧、深度图和上下文特征进行warp。在补充材料中提供了自适应warp层的更多细节以及Kernel估计网络的配置。

#### 帧合成(Frame synthesis)

为了生成最终的输出帧，我们构建了一个包含3个残差块的帧合成网络。我们将被warp的输入帧、被warp的深度图、被warp的上下文特征、映射流和插值Kernel连接起来，作为帧合成网络的输入。我们线性混合两个被warp的帧，并强制网络预测ground-truth帧和混合帧之间的残差。我们注意到，被warp的帧已经通过光流对齐了。因此，帧合成网络的重点是增强细节，使输出帧看起来更加清晰。我们在补充材料中提供了帧合成网络的详细配置信息。

### 实施细节(Implementation Details)

损失函数

我们用表示合成帧，用表示ground-truth帧。我们通过最优化以下损失函数来训练我们所提出的网络。

是Charbonnier惩罚函数(*P. Charbonnier, L. Blanc-Feraud, G. Aubert,and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In ICIP, 1994.)*我们一般将设置为1e-6

训练集

我们使用Vimeo90K数据集来训练我们的模型。Vimeo90K数据集有51312个三元组(triplet

)用于训练，每个三元组包含3个连续的视频帧，分辨率为256×448像素。我们训练我们的网络来预测每一个三元组的中间帧(即，t = 0.5)。在测试时间内，我们的模型能够为任意 t ∈ [0, 1] 生成任意中间帧。我们通过水平和垂直翻转以及颠倒三元组的时间顺序来对训练数据进行数据增强。

训练策略

我们使用AdaMax 来优化所提出的网络。我们将β1和β2设置为0.9和0.999，batchsize为2。kernel估计、上下文提取和帧合成网络的初始学习率设置为1e-4。由于流估计和深度估计网络是从预训练模型初始化的，因此我们使用较小的学习率，分别为1e-6和1e-7。我们训练整个模型一共30个epochs，然后将每个网络的学习率降低0.2倍，并对整个模型进行额外10个epochs的微调。

我们在一张NVIDIA Titan X (Pascal) GPU卡上训练我们的模型，大约需要5天收敛。

## 实验结果

在本节中，我们首先介绍用于评估的数据集。然后，我们进行消融研究，以分析我们所提出的深度感知流映射和分层上下文特征的贡献。然后，我们将提出的模型与最先进的帧插值算法进行比较。最后，我们讨论了我们方法的局限性和未来工作。

### 评估数据集和指标

我们在具有不同图像分辨率的多个视频数据集上评估了所提出的算法。

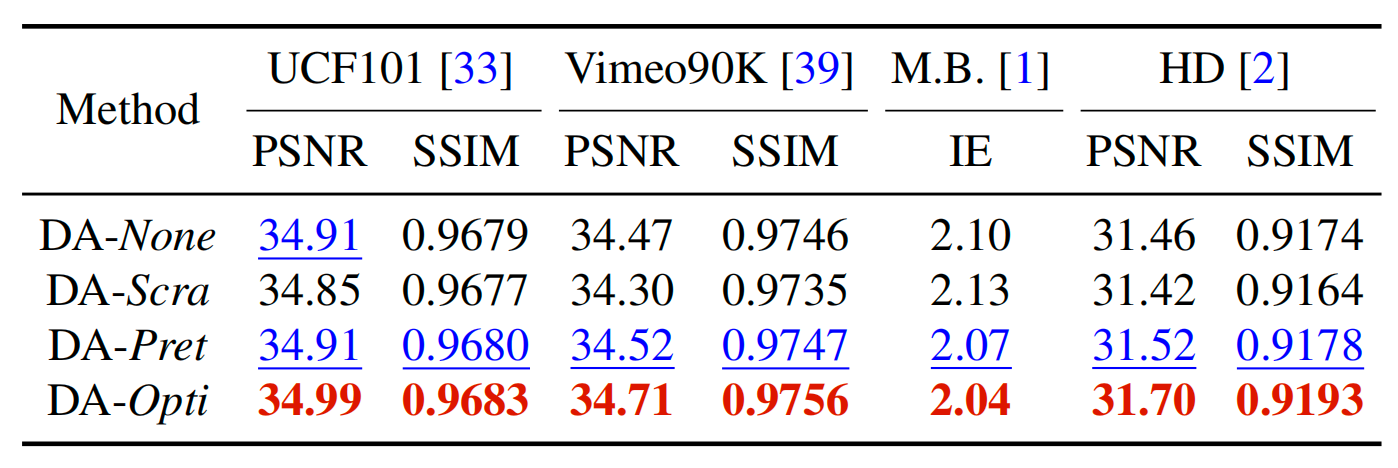


表 1 深度感知（DA）流投影分析。M.B.是Middlebury数据集其他集合的缩写。所提出的模型（DA-Opti）与其他变体相比有了实质性的改进。

**Middlebury**

Middlebury被广泛用于评估视频帧插值方法。有两个子集。训练集提供ground-truth中间帧，而评估集隐藏ground-truth，可以通过将结果上传到相关网站进行评估。此数据集中的图像分辨率约为640×480像素。

**Vimeo90K**

Vimeo90K数据集的测试集中有3,782个三元组。此数据集中的图像分辨率为448×256像素。

...

### 模型分析

我们分析了所提出的模型中两个关键组件的贡献：深度感知流映射层和可学习的上下文分层特征。

**深度感知映射**

为了分析我们的深度感知流映射层的有效性，我们训练了以下变体（DA是深度感知的缩写）：

DA-None：我们删除了深度估计网络，并使用简单的平均值计算来汇聚流映射层中的流。

DA-Scra：我们从头开始初始化深度估计网络，并针对整个模型进行优化。

DA-Pret：我们从(*Z. Li and N. Snavely. Megadepth: Learning single-view*

*depth prediction from internet photos. In CVPR, 2018.*)的预训练模型中初始化深度估计网络，但冻结了参数。

DA-Opti：我们从(Z. Li and N. Snavely. Megadepth: Learning single-view

depth prediction from internet photos. In CVPR, 2018.)的预训练模型中初始化深度估计网络，并与整个模型共同优化。

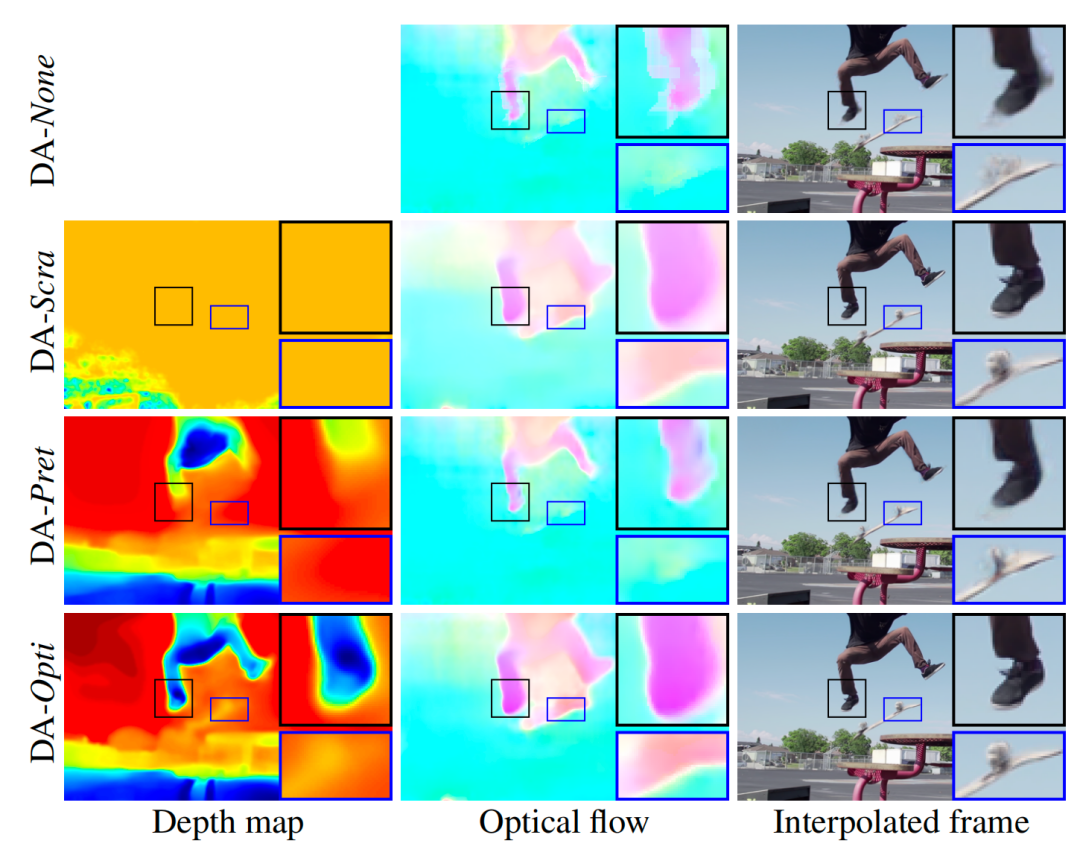


图 5 深度感知流映射的作用。 DAScra模型无法学习任何有意义的深度信息。DA-Pret模型从预训练模型初始化深度估计网络，并为帧插值生成清晰的运动边界。DA-Opti模型进一步优化了深度图，并生成更清晰的边缘和形状。

我们在表1中显示了上述模型的定量结果并且提供图5中深度、流动和插值帧的可视化。首先，DA-Scra模型的性能比DA-None模型差。如图5第二行所示，DA-Scra模型无法从随机初始化中学习任何有意义的深度信息。当从预训练的深度模型初始化时，DA-Pret模型表现出实质性的性能提升，并产生具有明确运动边界的流。在共同优化整个网络后，DA-Opti模型进一步改进了深度图，例如男人的腿，并为插值框架中的鞋子和滑板生成更锐利的边缘。分析表明，我们所提出的模型能够有效地利用深度信息来生成高质量的结果。

**Learned hierarchical context**

在我们提出的网络中，我们将上下文特征用作帧合成网络的一个输入。我们分析了不同上下文特征的贡献，包括预训练的conv1特征(PCF)、学习的conv1特征(LCF)和学习的层次特征(LHF)。此外，我们还将深度图(D)视为额外的上下文特征。

我们在表2中展示定量结果并且比较了图6中插值的图像。在不使用任何的上下文信息时，模型的表现将不尽人意并且生成较为模糊的结果。通过引入上下文特征，比如预训练的conv1特征或深度图，性能将大大提升。

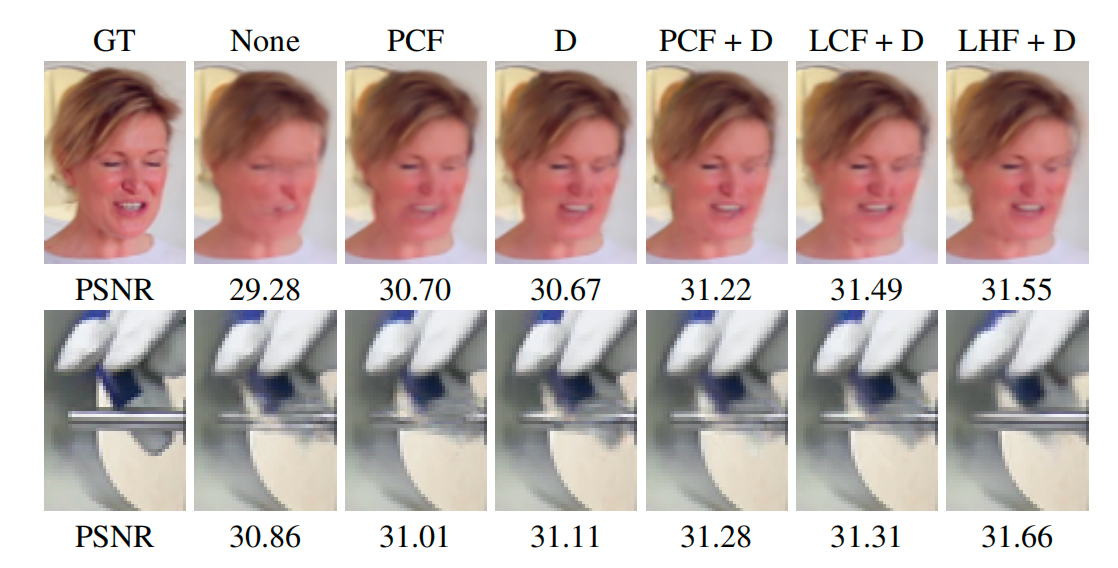


图 6 引入上下文特征的效果。我们所提出的模型使用学习分层特征（LHF）和深度图（D）进行帧合成，从而产生更清晰、更清晰的内容。

### 与当前最佳效果的算法比较