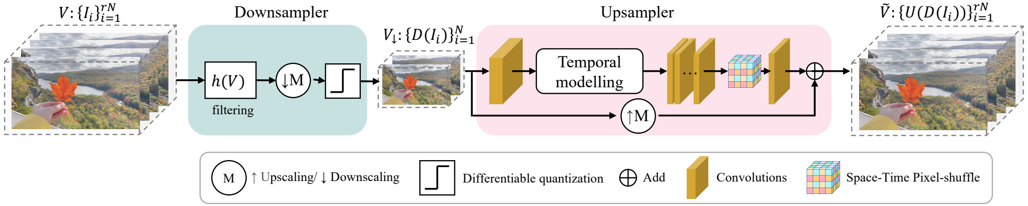
Learning Spatio-Temporal Downsampling for Effective Video Upscaling

Xiaoyu Xiang[1] Yapeng Tian[2] Vijay Rengarajan[1] Lucas Young[1] Bo Zhu[1] Rakesh Ranjan[1]
[1]Meta Reality Labs [2]University of Texas at Dallas

ECCV TEL AVIV 2022



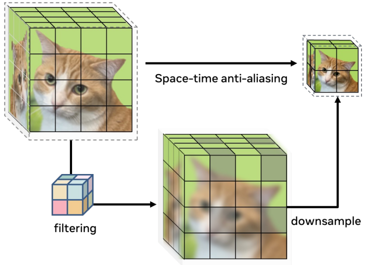
Motivation

Given an image, how to downsample it into a smaller one?

- If we directly take one pixel from a region, then the result will be with obvious jigsaws due to aliasing.
- To avoid these artifacts, we need anti-aliasing filters before downsampling, like bicubic, and gaussian filters.

Now given a video sequence of many images, how do we downsample it?

- We regard video as a 3D xyt volume, and propose to use a 3D space-time anti-aliasing filter to downsample it.
- Like what we usually do for an image, we first generate a filtered cube, and then downsample it by striding.
- In this way, we can get a better downsampled output without aliasing in space and time.



Framework

To better retain and recover spatio-temporal details, we design a framework that jointly learns a downsampler and an upsampler that effectively captures and reconstructs high-frequency details in both space and time.

The **downsampler** includes 3 parts:

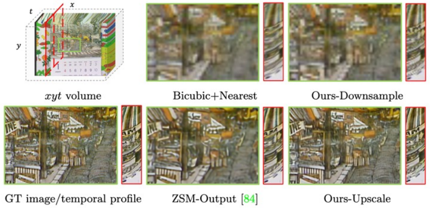
- A space-time anti-aliasing filter
- A downsampling operation
- A differentiable quantization layer

For the **upsampler**:

- We adopt 3D convolutions as the basic building block. However, naïve 3D convolution is insufficient.
- So we propose to enhance the temporal correspondences with a deformable temporal modeling network.
- We devise a space-time pixelshuffle module. It rearranges the feature channel elements into both space and time dimensions.

Why our method is better?

Look at this video cube:



The result from the commonly used bicubic downsampling and skip frames is not optimal for reconstruction.

While our learned downsampler can keep better temporal textures.

Hence, our reconstruction result has richer details and better motion patterns than previous SOTA method.

Experiments

Downsampler	Params/M	GFLOPs/MP	PSNR	SSIM
CAR [66]	9.896	2305.77	35.96	0.9400
PASA [94]	0.003	6.144	35.37	0.9524
Ours	**0.002**	**0.081**	**37.35**	**0.9629**

Tab 1. Comparison of spatial downsamplers (1×t, 4×s).

Time	Space	PSNR	SSIM
Nearest	Bicubic	28.88	0.9073
Gaussian		37.44	0.9679
STAA_{no}		39.44	0.9775
STAA_{soft}		40.40	0.9812
STAA_{quant}		40.42	0.9811
STAA_{ada}		38.13	0.9720

Tab 2. Quantitative comparison of downsampling filters (2 × t, 2 × s). The best two results are highlighted in red and blue, respectively.

Application

- Efficient video storage
- Blurry frame reconstruction
- Arbitrary frame rate conversion