Denoising with Kernel Prediction and Asymmetric Loss Functions

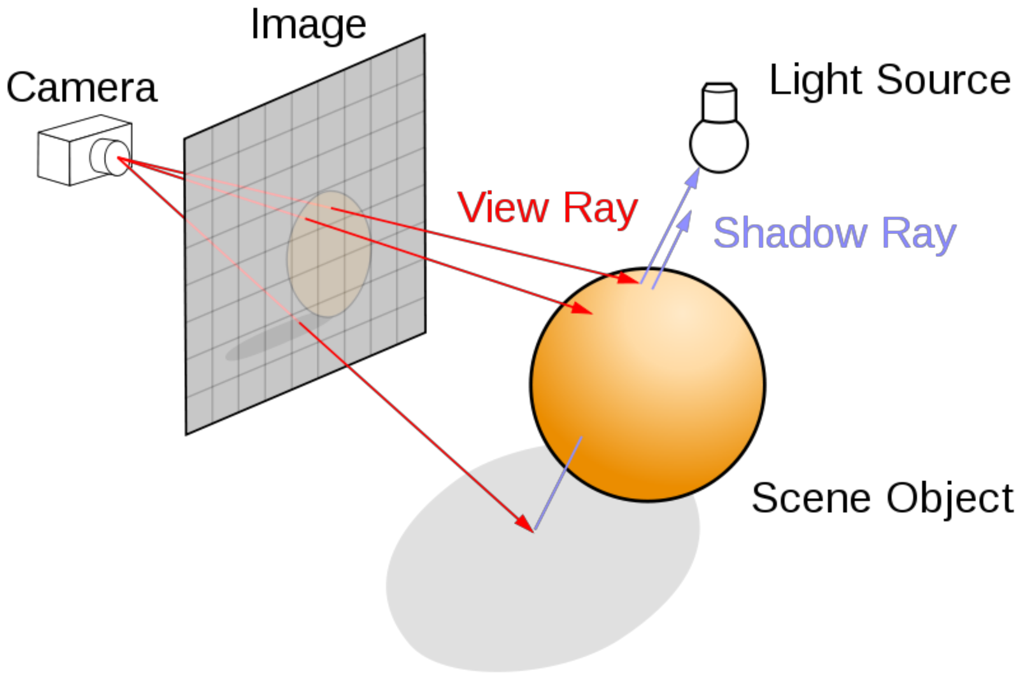
Minmin GE

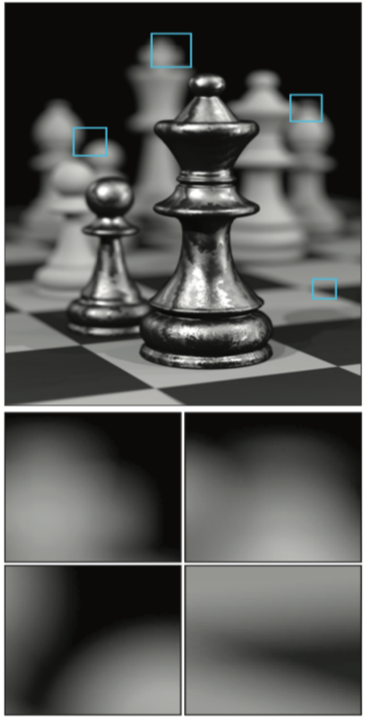
Plan

1. Background:
   - MC rendering
   - Denoising methods
2. Denoising with kernel prediction (Statistic method)
   - Single frame denoiser
   - Temporal stability
   - Multi-scale architecture
3. Asymmetric loss function

Background:
MC rendering & Denoising methods

MC rendering





Reference (1024 samples/pixel)

Denoising methods

Local smoothing filter:

- A continuous image is usually interpreted as the Shannon interpolation of a discrete grid of samples.

Other methods:

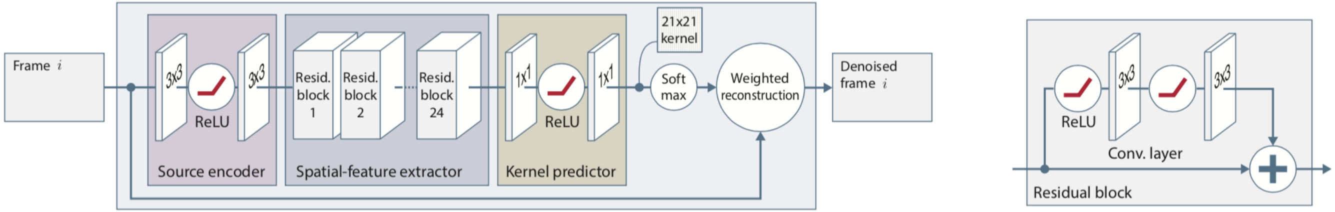
- Use different kernels
- Can be categorized depending on the polynomial order

Neighborhood filter:

- take the average of values of the neighbor pixels
- neighborhood is defined as a group of pixels which has the closest values
- Non local method

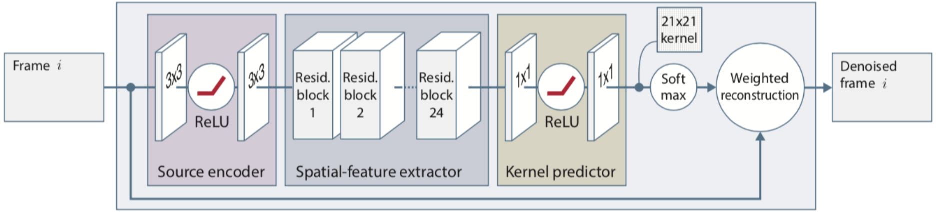
Denoising with kernel prediction

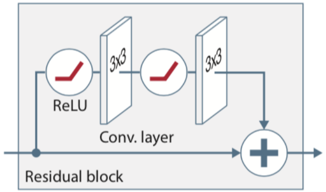
Single frame denoiser



- The features of the input may depend strongly on the renderer
- A source encoder is used to extract the input features unique to a certain renderer.
- Trained independently of the following layers

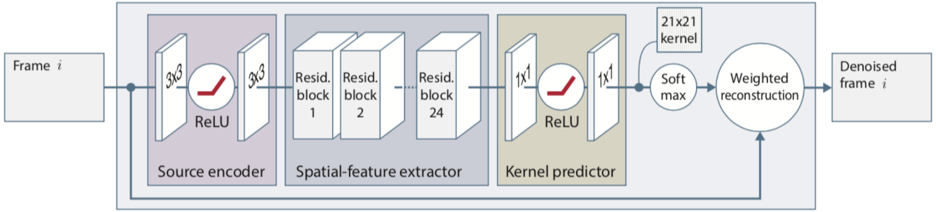
Single frame denoiser

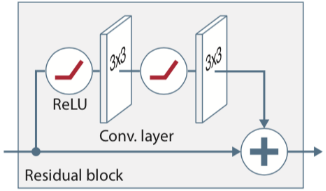




- The core of the denoiser is a kernel predicting CNN using 3 residual blocks
- Residual blocks can improve the performance significantly without introducing optimization instabilities

Single frame denoiser



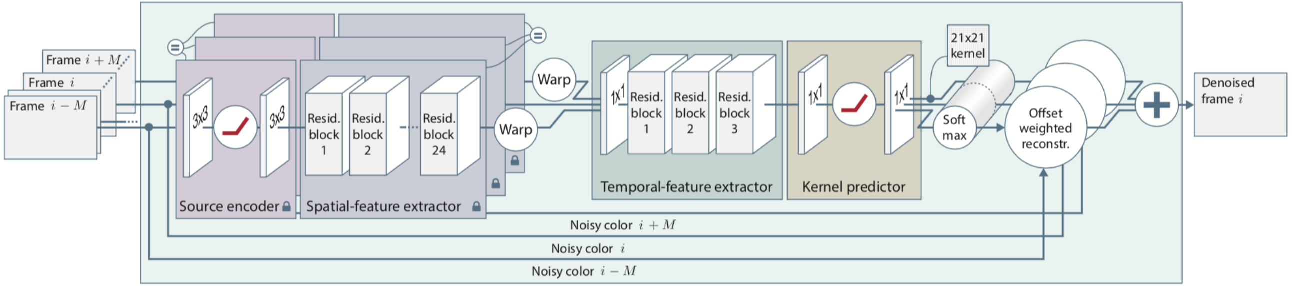


- The denoised color reconstructed as a linear combination of input colors in a *kxk* neighborhood around it

Flickering

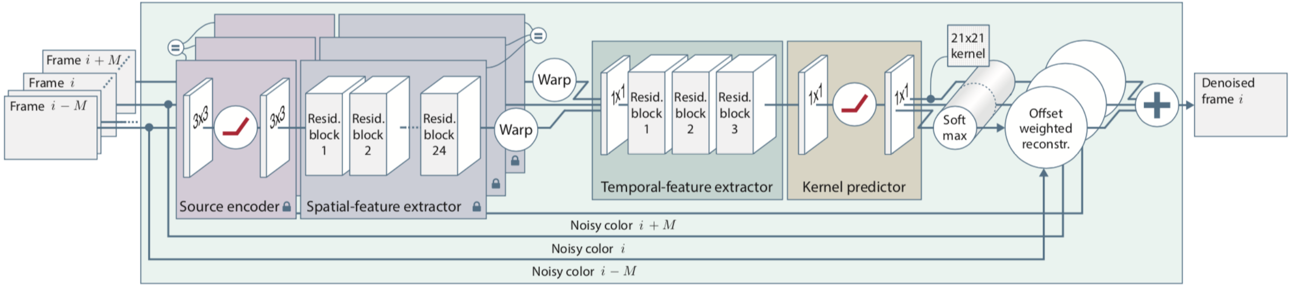
The single frame denoiser can produce severe artifacts when executed on a sequence of frames independently since each denoised frame may be “wrong” in a slightly different way

Temporal stability



- The extracted features of different frames are warped using motion vectors to match the time of the center frame

Temporal stability

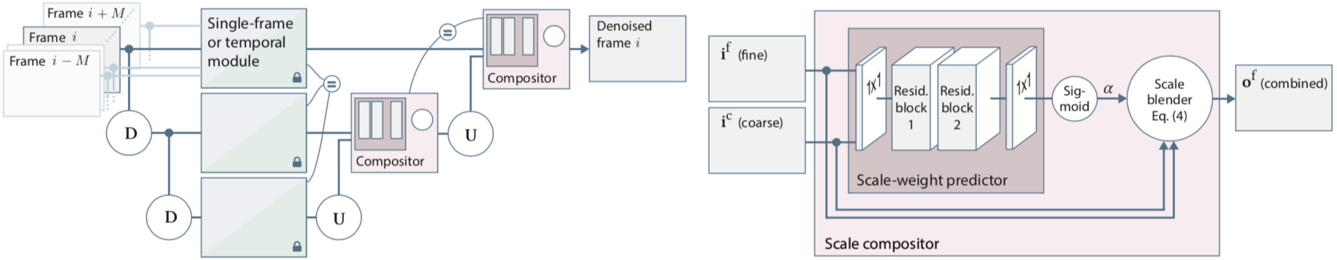


- These warped features are the concatenated and fed into a temporal feature extractor
- A kernel predictor is then applied on the features

Residual noise

The algorithm is typically good at removing high frequency noise but it may leave low frequency noise.

Multi-scale architecture



- A three level pyramid is constructed by a uniform 2x2 down-sampling

Asymmetric loss function

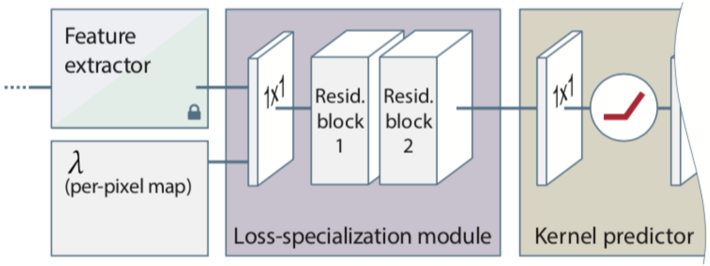
Asymmetric loss function

- Goal:
    - Under some circumstance, it may be desirable to retain some noise
    - An artistic decision
- Solution: Use an asymmetric loss function

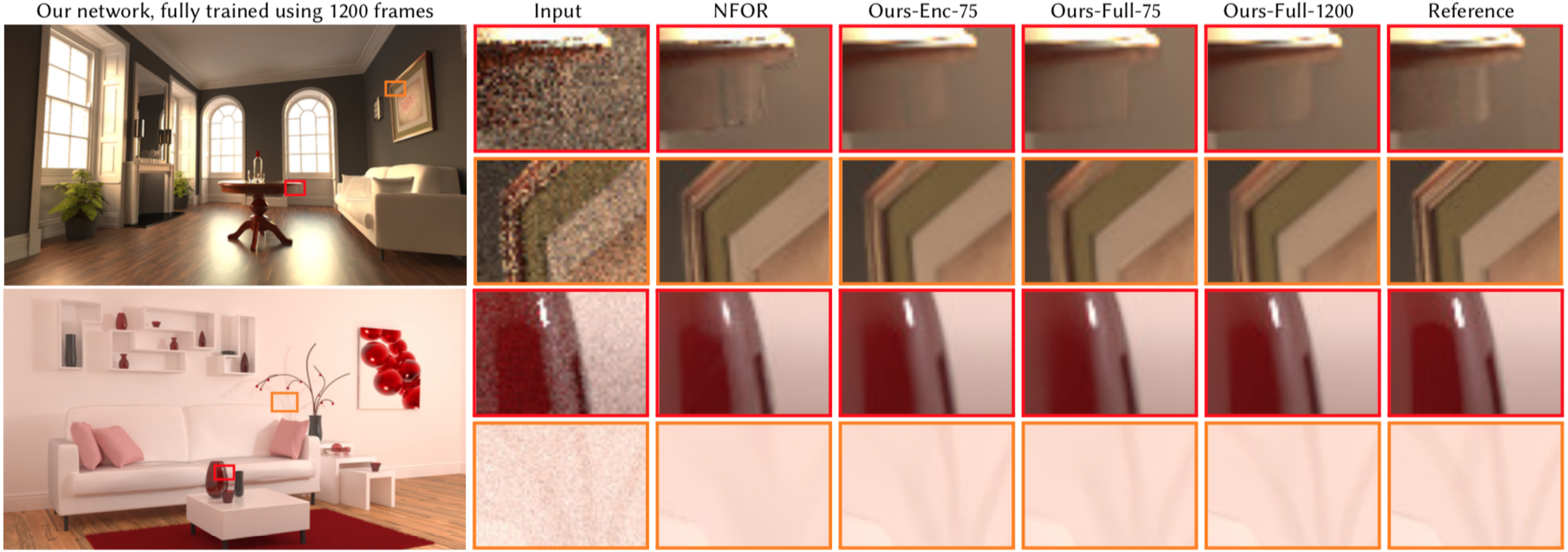
$$\ell'_\lambda(\mathbf{d}, \mathbf{r}, \mathbf{c}) = \ell(\mathbf{d}, \mathbf{r}) \cdot (1 + (\lambda - 1)H((\mathbf{d} - \mathbf{r})(\mathbf{r} - \mathbf{c})))$$

where d is the denoised color, r is the reference color and c is the input color
H returns 1 when the argument is positive or 0 and 0 otherwise

Asymmetric loss function



- take λ, like the features extracted, as the input of the loss-specialization module
- the loss-specialization module output then a representation of the features and λ to the kernel predictor



Result (provided by Pixel)