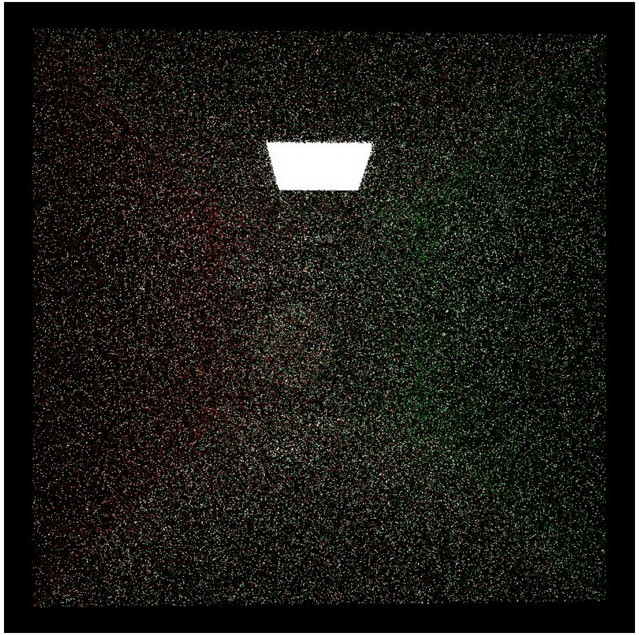
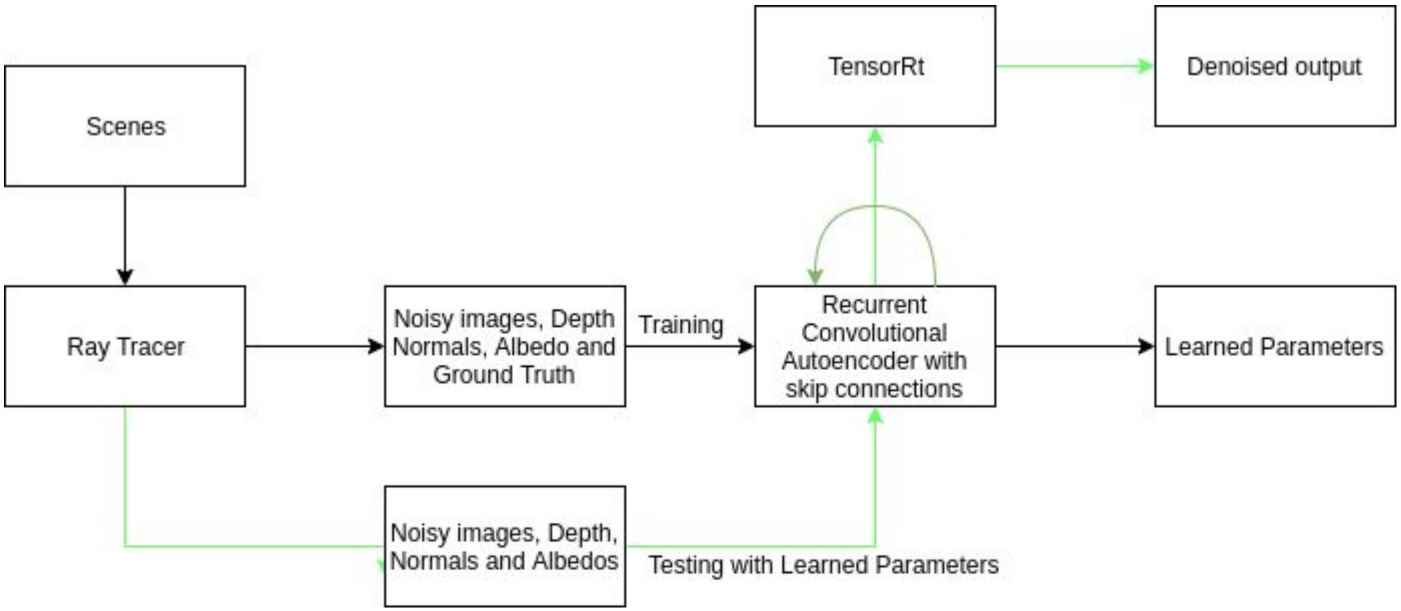
Another Image (AI) Denoiser

Vaibhav Arcot & Dewang Sultania (Milestone 2)

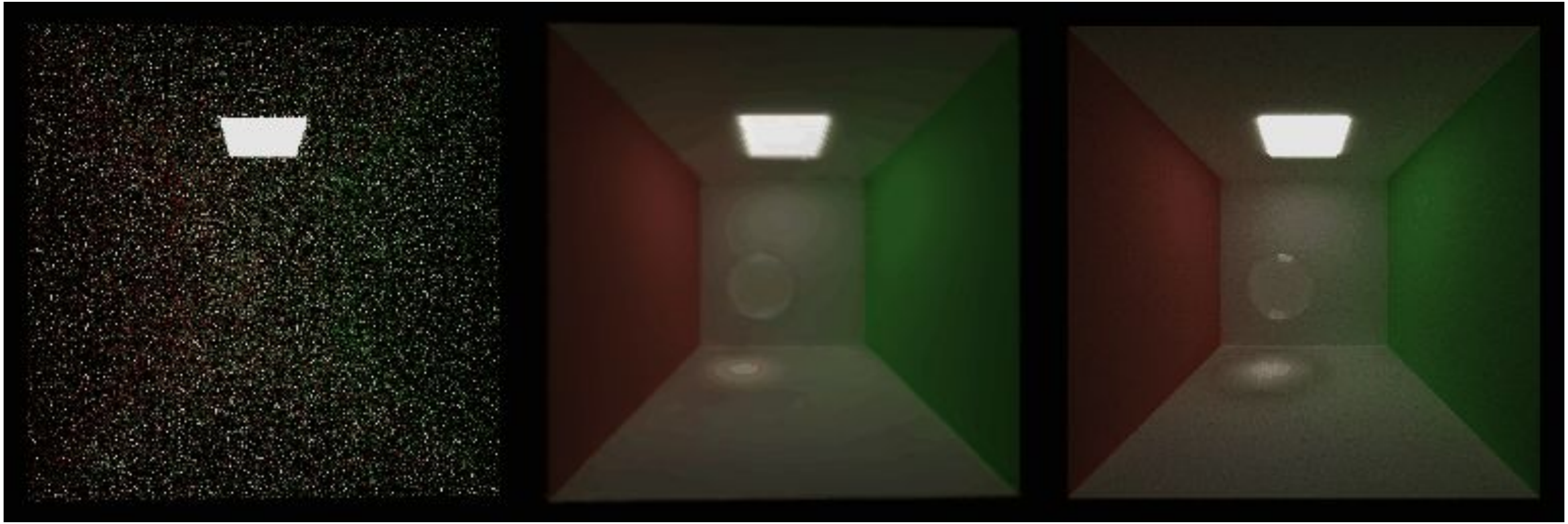
Image sequences



Workflow


TensorRt
Denoised output
Scenes
Ray Tracer
Noisy images, Depth Normals, Albedo and Ground Truth
Training
Recurrent Convolutional Autoencoder with skip connections
Learned Parameters
Noisy images, Depth, Normals and Albedos
Testing with Learned Parameters

Results

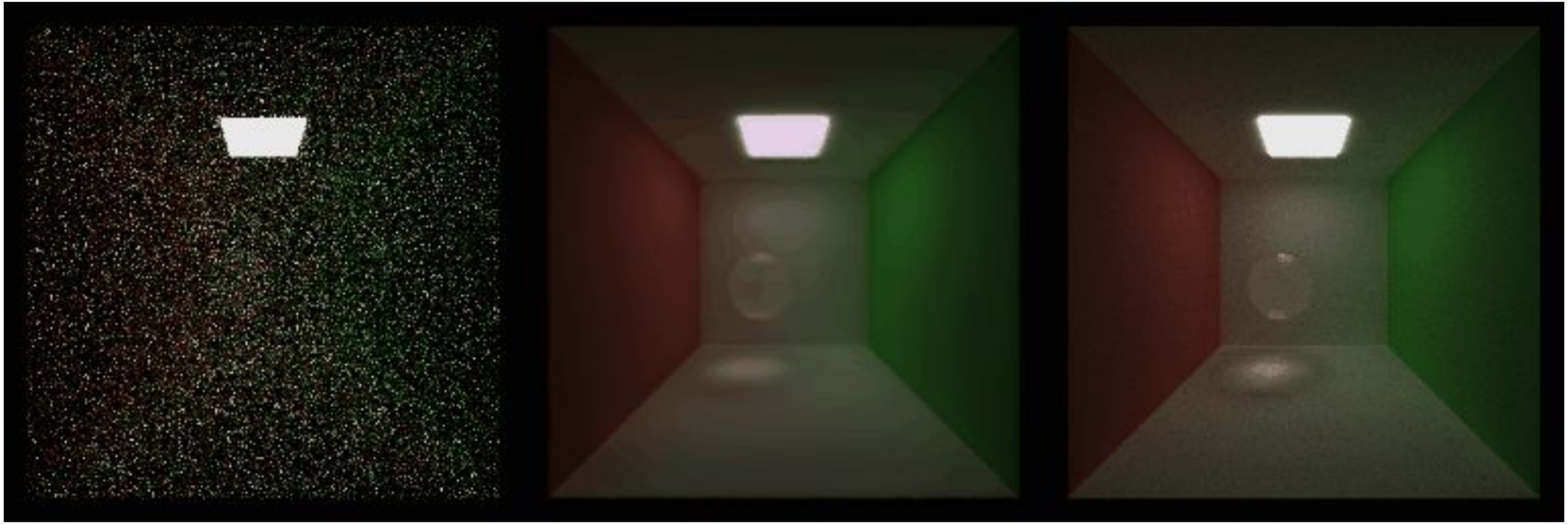


1SPP

Denoised

2500 SPP

Results



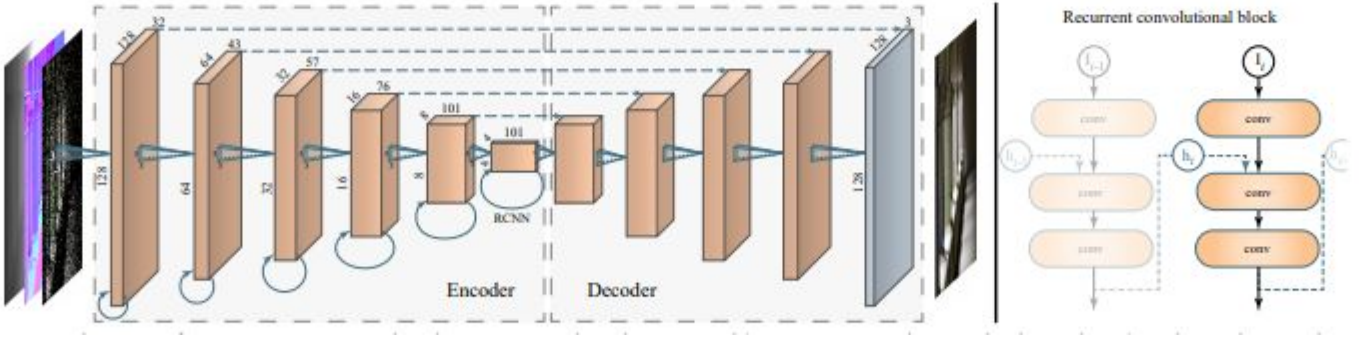
1SPP

Denoised

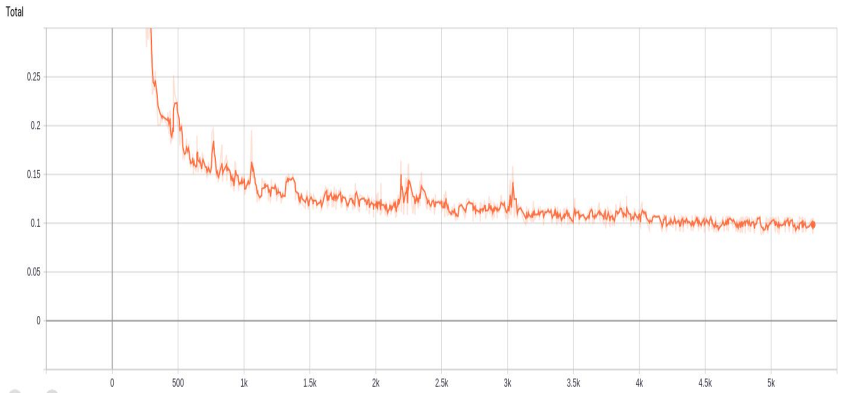
2500 SPP

https://research.nvidia.com/sites/default/files/publications/dnn_denoise_author.pdf

Architecture - Recurrent Convolutional Autoencoder with Skip Connections

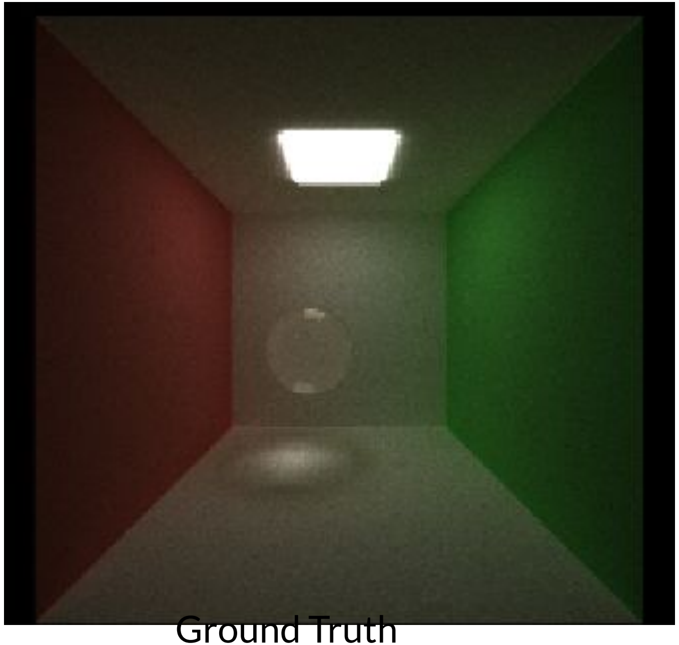


Loss Function



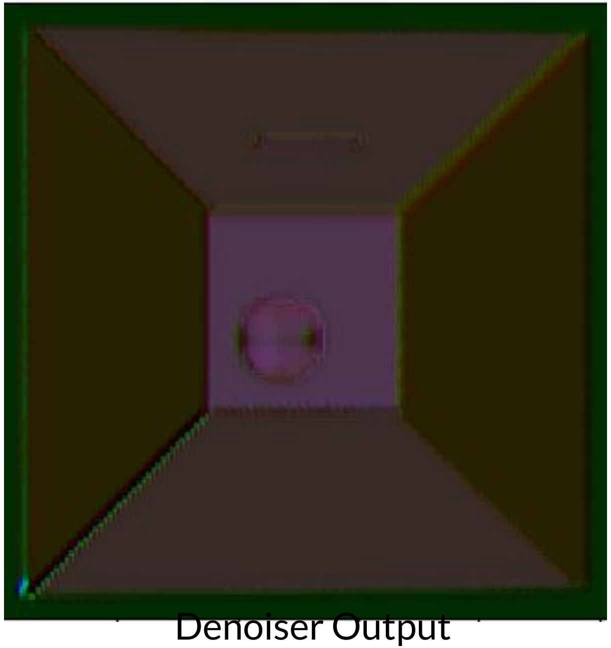
- Loss function is a weighted avg of
  - Spatial L1 Loss
  - High Frequency Error Norm
  - Temporal Loss
- We use a gaussian curve to modulate the weights on each loss.

Spatial L1 Loss



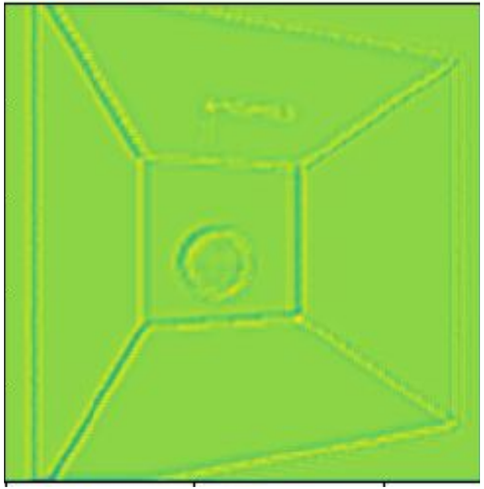
Ground Truth

Reduces difference b/w these two



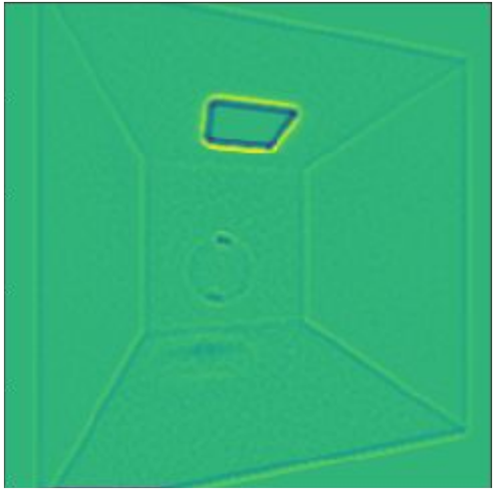
Denoiser Output

High Frequency Error Norm



Ground Truth

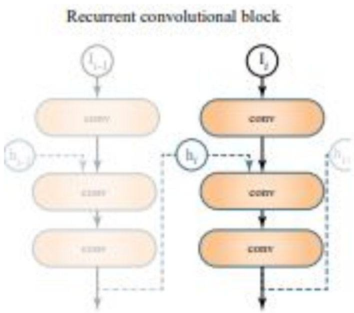
Helps penalise finer details like edges using LoG



Denoiser Output

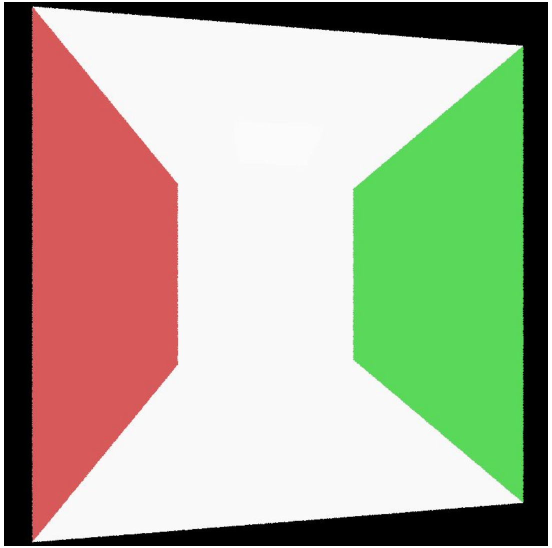
Temporal Loss

- Reduces flickering between frames

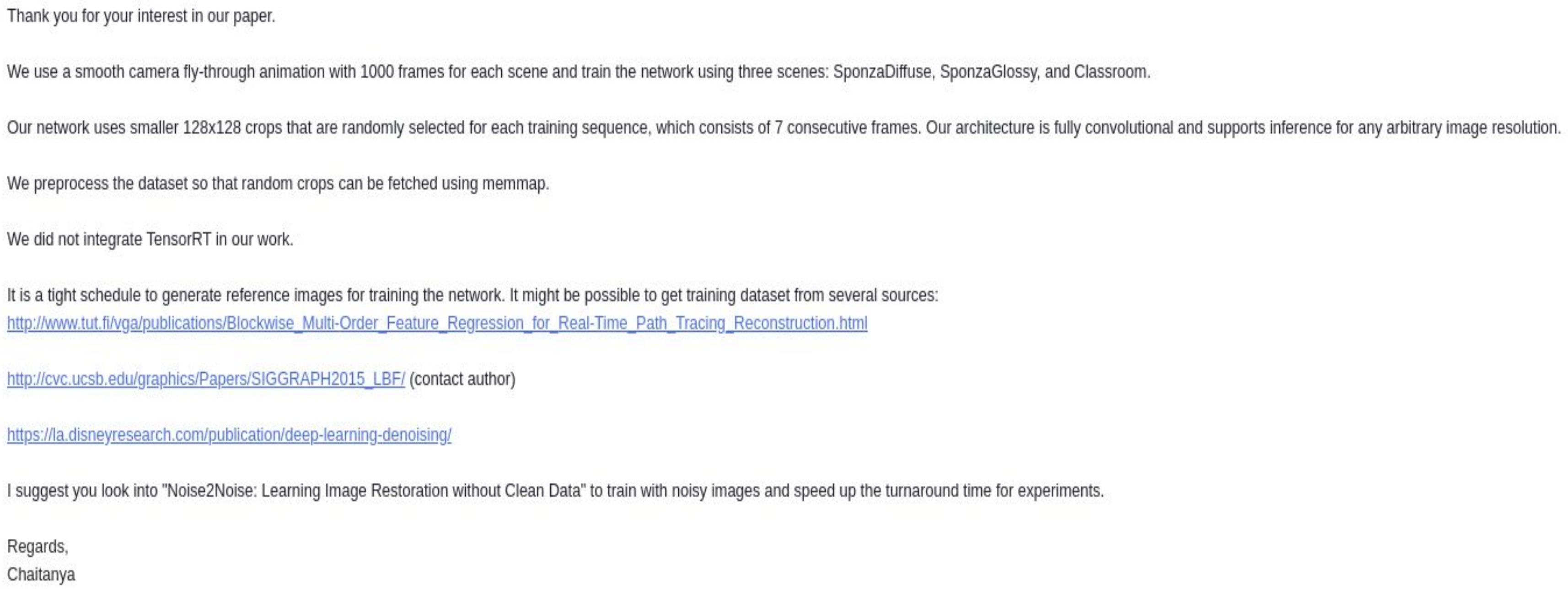


Additional Contributions to improve results

- As suggested by Alan Galvan, added albedo buffer as an input to the network to help it learn colors better.
- Increases the number of input channels to 10.



Author Reply - Basically next steps



Thank you for your interest in our paper.

We use a smooth camera fly-through animation with 1000 frames for each scene and train the network using three scenes: SponzaDiffuse, SponzaGlossy, and Classroom.

Our network uses smaller 128x128 crops that are randomly selected for each training sequence, which consists of 7 consecutive frames. Our architecture is fully convolutional and supports inference for any arbitrary image resolution.

We preprocess the dataset so that random crops can be fetched using memmap.

We did not integrate TensorRT in our work.

It is a tight schedule to generate reference images for training the network. It might be possible to get training dataset from several sources:
http://www.tut.fi/vga/publications/Blockwise_Multi-Order_Feature_Regression_for_Real-Time_Path_Tracing_Reconstruction.html

http://cvc.ucsb.edu/graphics/Papers/SIGGRAPH2015_LBF/ (contact author)

https://la.disneyresearch.com/publication/deep-learning-denoising/

I suggest you look into "Noise2Noise: Learning Image Restoration without Clean Data" to train with noisy images and speed up the turnaround time for experiments.

Regards,
Chaitanya

More Next Steps

- Generate more scenes and training data.
- Implement KD-tree to speed up data generation.
- Train the network on the new images.
- Speed up inference using TensorRT.
- Tile the input image to achieve larger output resolution.