

# Flow Based Image Abstraction

Shreyas Goenka, 17d070037

Harekrishna Rathod

Ayushraj Melhotra

# Algorithm:

1. Gaussian Smoothing
2. ETF (Edge Tangent Flow)
3. FDoG (Flow Difference of Gaussian)
4. FBL (Flow Bilateral Filtering)
5. Luminisence Quantization

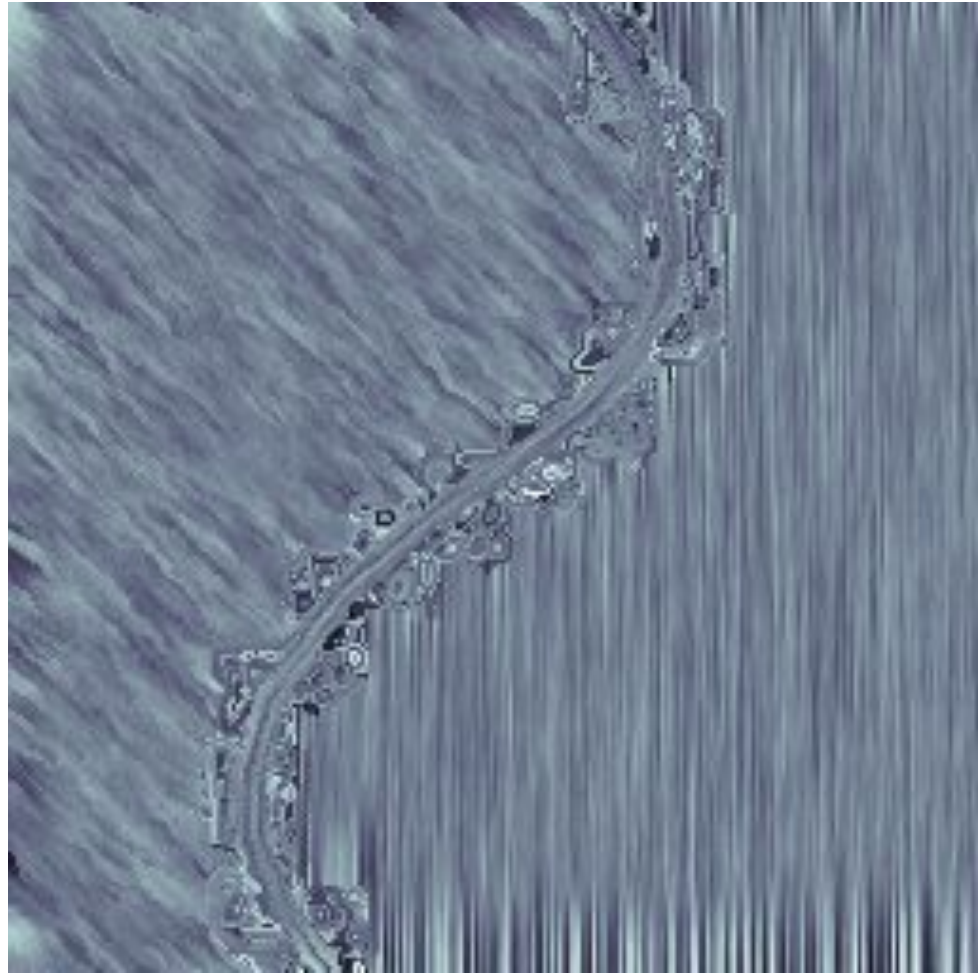
# ETF

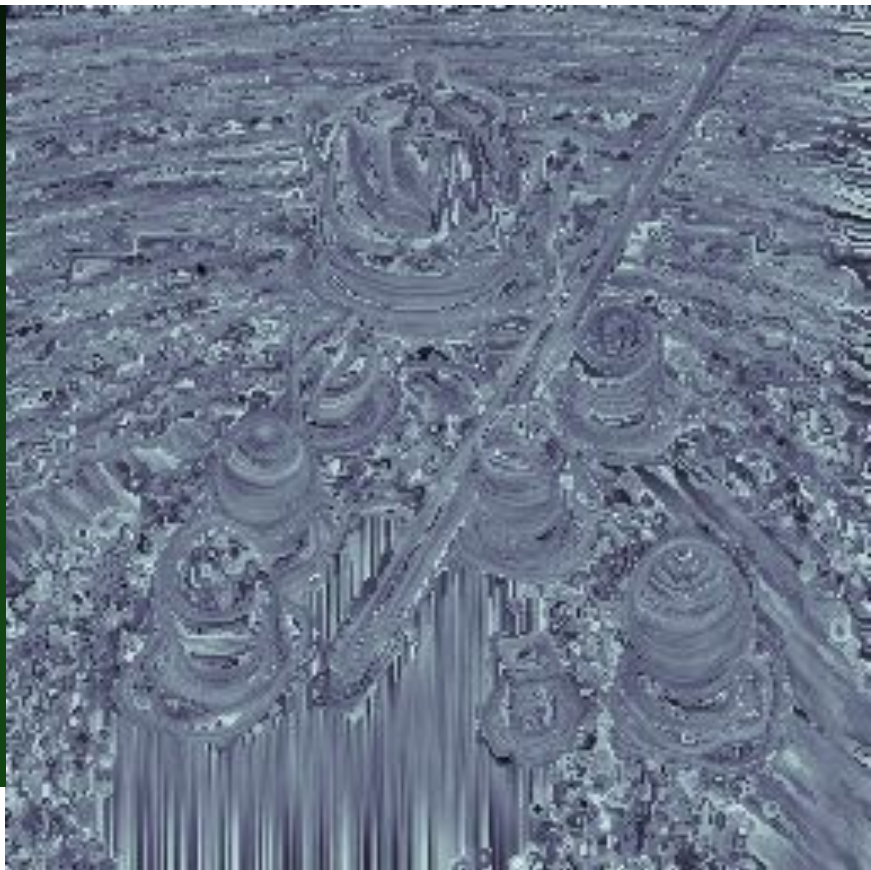
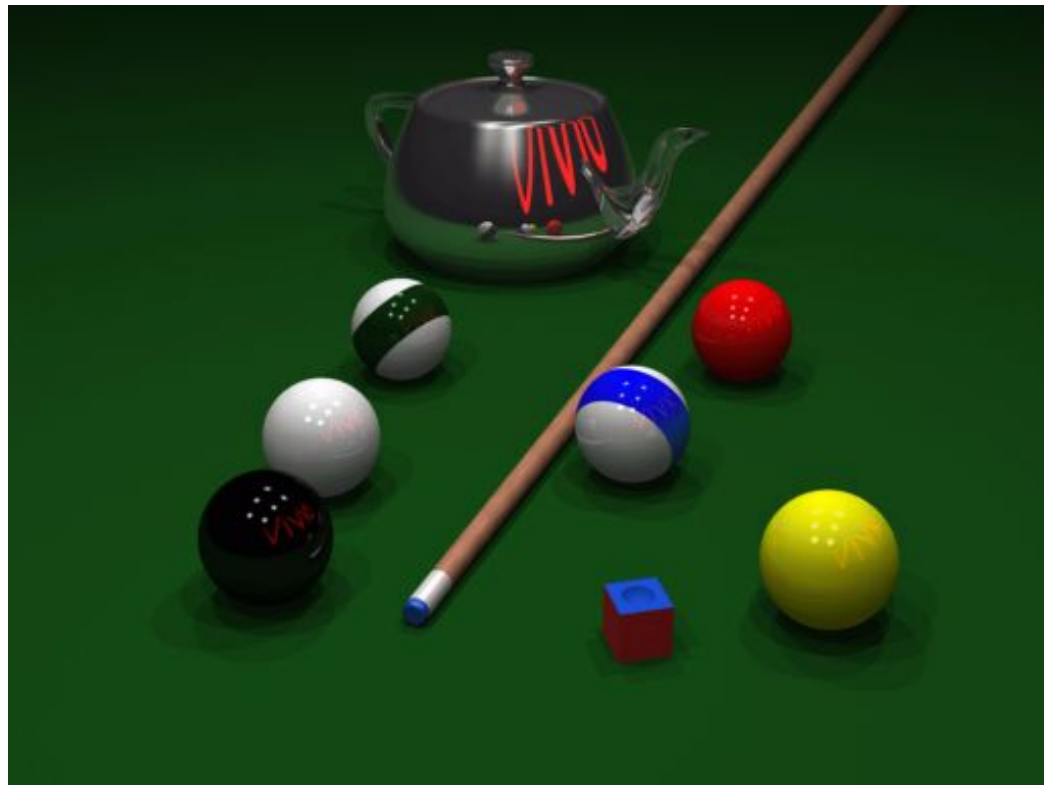
Iteratively calculates the tangent flow for an image, based on the paper by Henry Kang: [Coherent Line Drawing](#)

Visualized using Line Integral Convolution.

We implemented the 2D version with an complexity of  $O(n \cdot \mu \cdot \mu)$  where  $n$  is the number of pixels and  $\mu \times \mu$  square around the pixel is considered in the calculations.

We used two iterations of the ETF algorithm.





# FDoG

DoG along the gradient to detect edges.

Gaussian Smoothing along the tangent flow

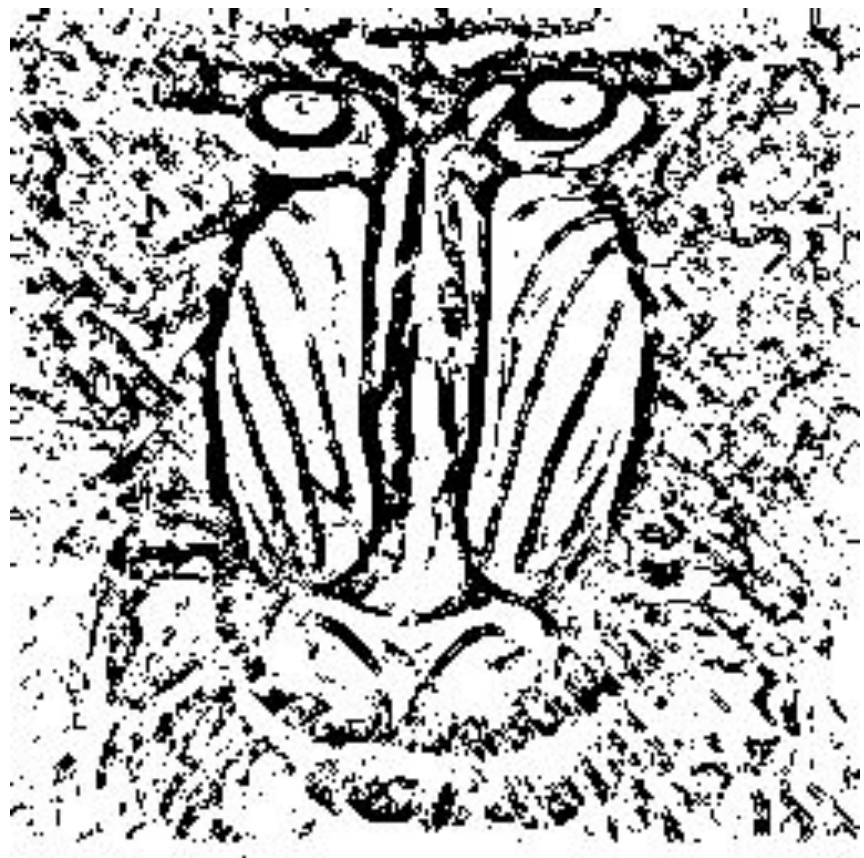
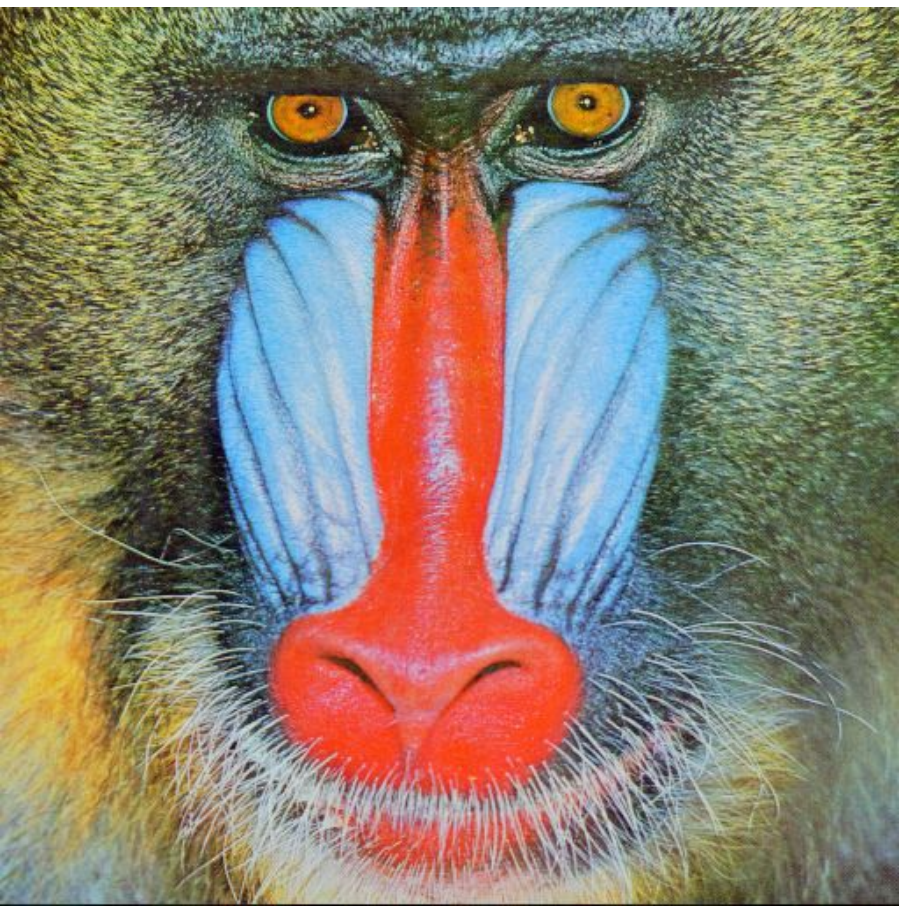
Challenge: the hyperparameters from the paper were not suitable to work with. For the algorithm to work the sum of DoG discrete components should be closed to 0 (to remove constant areas from consideration). We used the smallest recommended  $p$  by the paper that is 0.97 and calculated that  $\sigma_s = \sigma_c * 1.05$  rather than  $\sigma_s = \sigma_c * 1.6$  recommended by the paper.

The complexity is  $O(n * \alpha * \beta)$

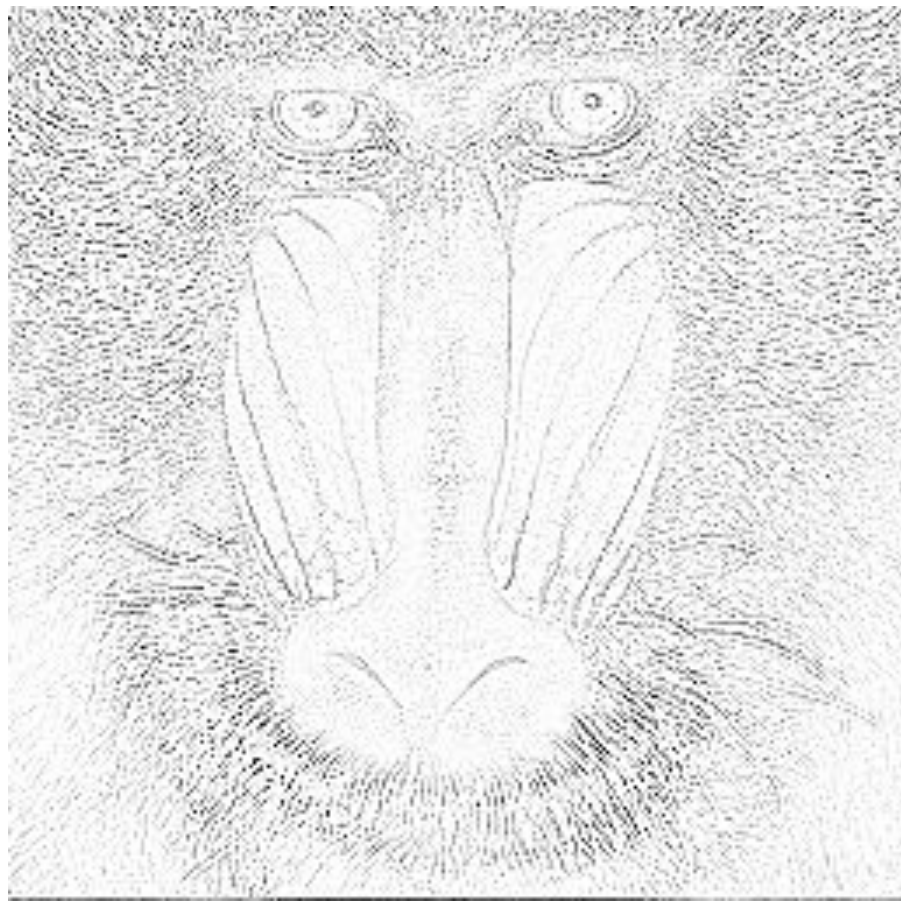
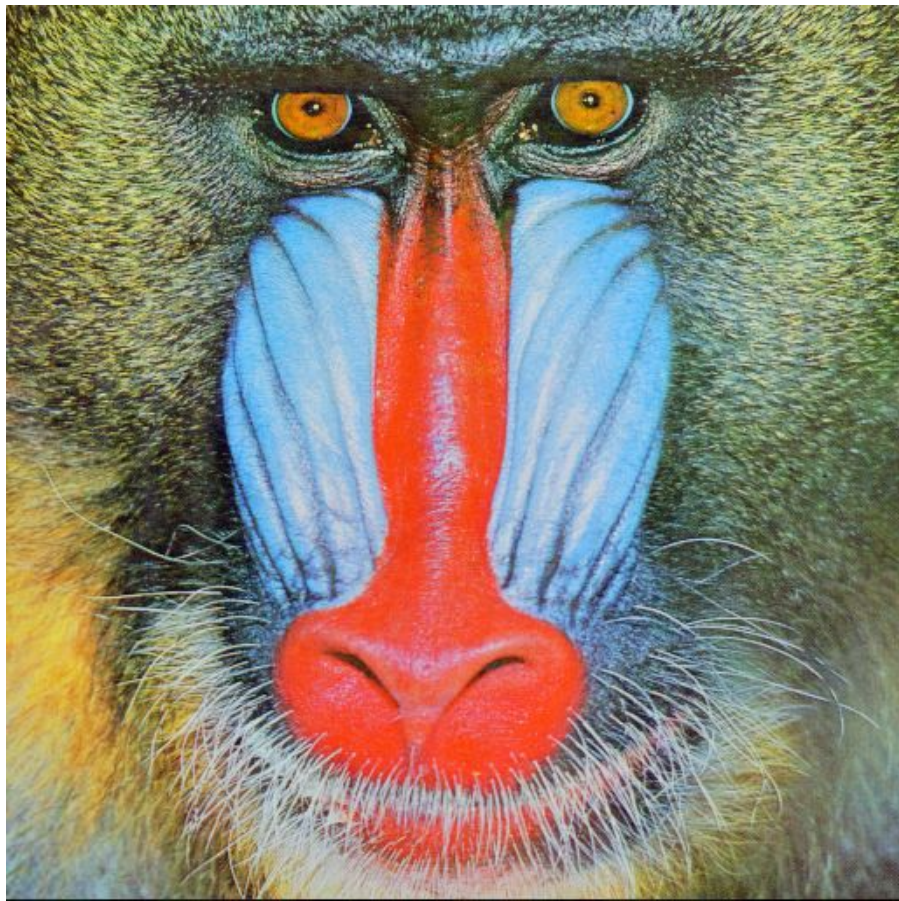


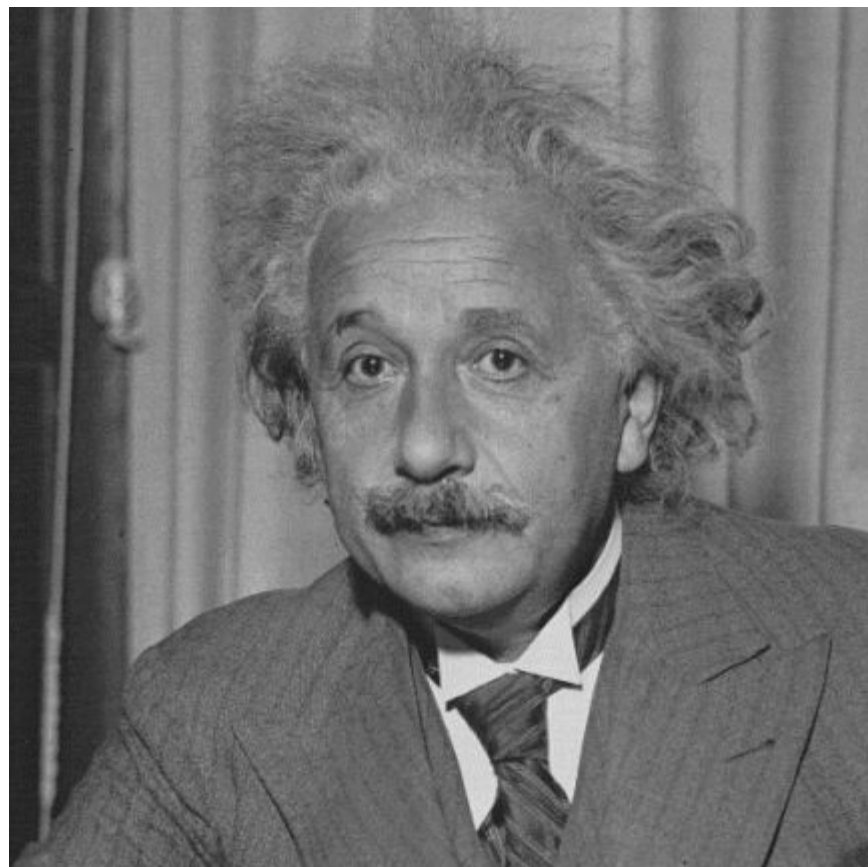


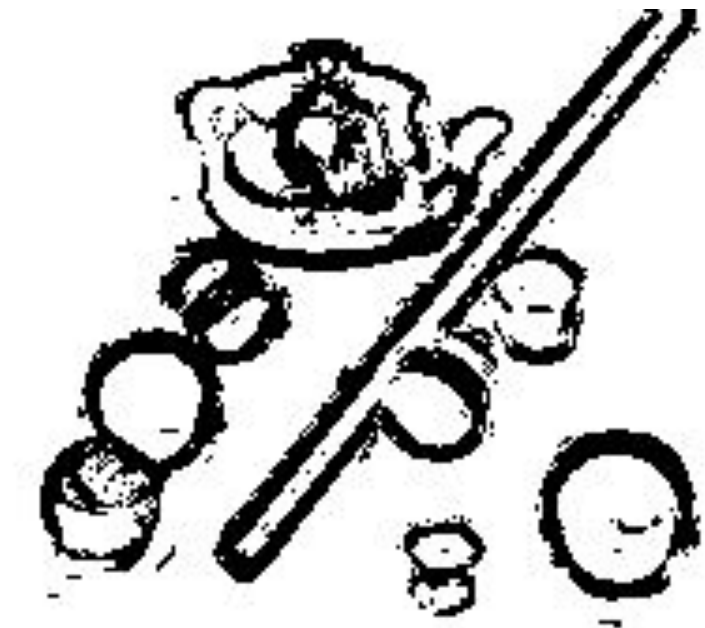


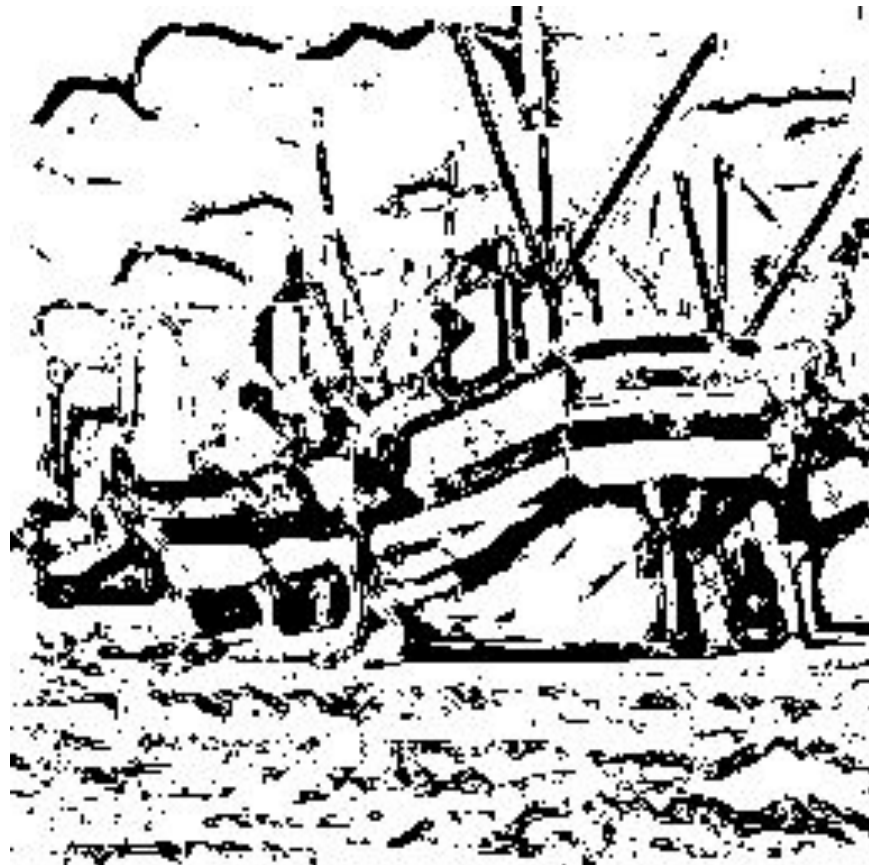












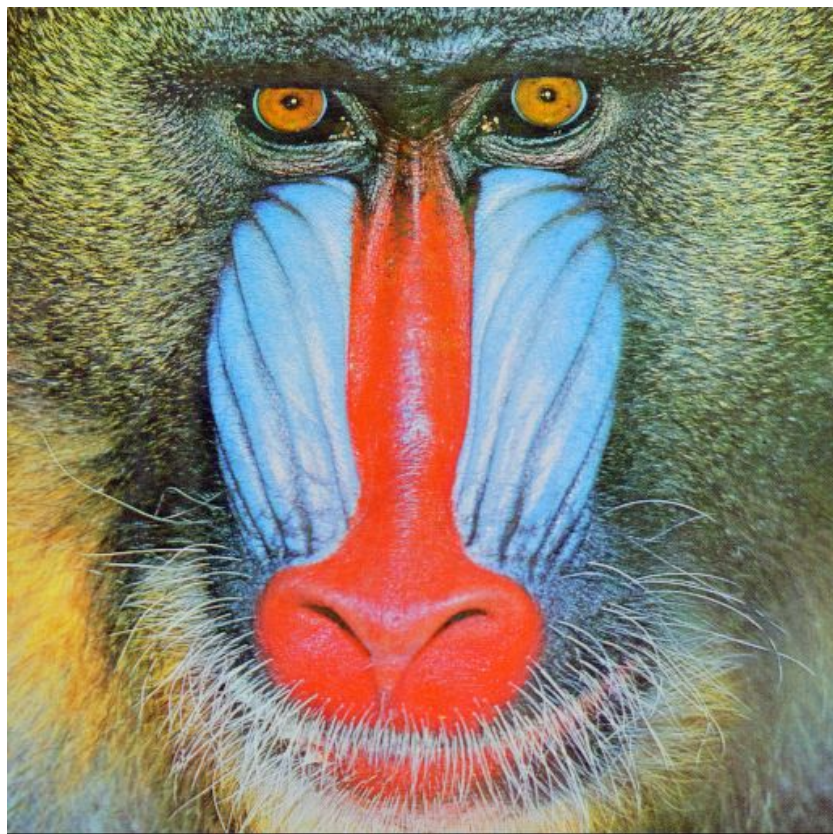
# FBL

Here rather than bilaterally filtering accross all direction, we only apply the filter in the direction of the tangent flow and the gradient flow.

We chose the standard deviation accross the tangent to be larger than across the gradient since the we dont want the edges to get blurrred. which was the the primary disadvantage of the original bilateral filter in this application.

The complexity is  $O(n \cdot \alpha \cdot \beta)$ , alpha and beta were taken to be 5 (ie 11 pixels in each direction).











# Luminence Quantization

We use a simple and naive quantization method where we quantize we divide the luminisence into 10 bins, thus having 10 direcete luminisence quantites in the image.

We will now display the final result images.





