Video Toonification

Madhur Sudarshan (17D070009)
Shubham Anand Jain (17D070010)
Mithilesh Vaidya (17D070011)

Motivation & Background

- Temporal consistency is lost in frame-to-frame toonification
- Idea to tackle presented in the paper “Video Tooning”, MSR
- Mean Shift Segmentation on Images with time as a dimension
- They mention that Mean shift is much better in maintaining temporal consistency compared to frame-to-frame toonification - we test this hypothesis

Benchmarking of results

- Toonification is a subjective process - no single benchmark
- However, we can compare jitter across processes, to check spatio - temporal consistency of pixels in a video
- We will compare two procedures:
  - Bilateral filtering + Canny Edge detection applied frame by frame
  - Temporal Mean Shift Filtering

Demonstration

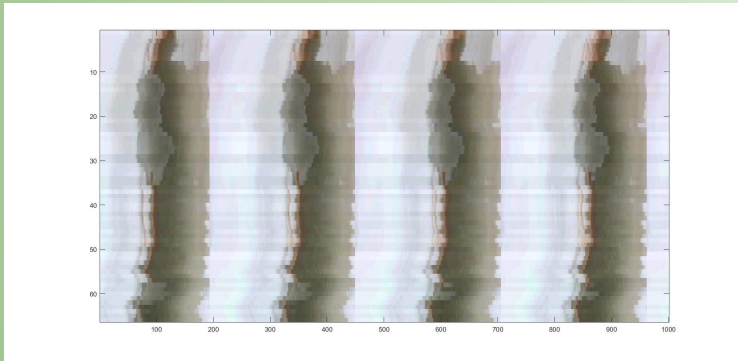
(We show the videos via screen-share)

For each video, we show Original and Filtered results

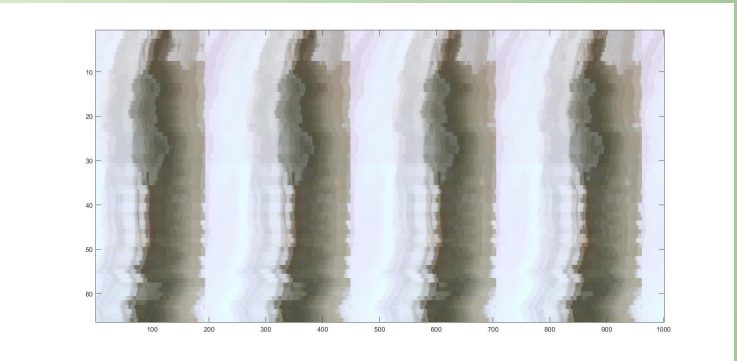
Link here

Comparison with Bilateral Filtering

- We note that the change in pixel colour across time is much lesser in mean-shift qualitatively compared to bilateral filtering



Bilateral Filtering



Mean Shift filtering

Implementation Specifics

- No manual intervention
- Scene Detection
- Canny Edge based boundaries
- Fixed Kernel widths as opposed to dynamic anisotropic kernel
- Downsampling Videos in time and frame size
- Grid search for optimal parameter selection

Inferences

- We observe that mean-shift significantly improves temporal consistency of toonified videos created
- Canny Edge detection based boundaries is akin to an artist using thick ink - the results are visually pleasing
- Manual Parameter tuning for Mean Shift is a significant problem

Future work

- Comparison with Flow-based models
- Speed up Mean Shift filtering
- Use local search or other methods for optimal parameter selection using results from the pixel-time curves
- Temporal consistency of edge detection

How to speed up Mean shift?

Ideas we had:

- Windowed Kernel (Reducing Search Space)
- Convert RGB to YCbCr and use properties of Y

Ideas from papers we saw:

- Overlapping windows (Implemented in “Video Tooning”)
- Other papers which can be used, eg. Gaussian KD-Tree for KNN

Questions?

→

Appendix - Bilateral Filtering

- Do bilateral filtering with empirically chosen sigma’s (perhaps by grid search)
- Use canny edge detection with a chosen threshold to sharpen edges
- Quantize the colors (we use a = 8)

Appendix - Scene Detection

- If a video has multiple scenes, Mean Shift Segmentation across the entire video will give incorrect results
- We implemented a very simple Scene Detection algorithm based on the RMSE difference between consecutive frames
- If the difference exceeds the moving average by a particular factor (which is tuned), we declare a scene boundary
- Mean Shift Segmentation is applied separately to each of the segments

Appendix - Mean Shift

- Normalized feature space to compare sigma
- Different sigma for space, time, intensity
- Scaled Features to simplify kernel based calculations
- Implemented in MATLAB
- Downsampled assuming input videos are 720p (1280 x 720 pixels) at 30 fps
- Takes upto 10-15 minutes for a 5 second long scene