Challenge 5 Report

Juan Reyes
University of California, Santa Barbara

Marco Chavez
University of California, Santa Barbara

***Abstract*—Rotoscoping is a widely used technique for creating animations from live-image/video. There are many techniques to implement digital rotoscoping. This report discusses an implementation that uses the sobel operator and bilateral filter in MATLAB.**

I. INTRODUCTION

Rotoscoping produces cartoon like animation by tracing over objects in frames creating sketches of each frame. Many of the digital rotoscoping techniques involve contrast adjustment, adaptive filtering, and edge processing. This report discusses a method that uses a sobel filter to outline the edges and a bilateral filter to blur the frames while preserving the edges.

Sobel filter is a combination of two filters to find the gradient of distances and intensities in an image. The result is an outline of where the image has high variation in light intensities. Since objects usually vary in light intensities from the background and other objects the result is a black and white sketch like frame of the original frame outlining objects in the frame.

Bilateral filter applies a blur using a weighted filter. This filter emphasizes edges by giving large weight to pixel to similar value as the pixel in question and low weights to pixels that are not similar.

Both these filters calculate intensities of an image. The sobel filter is used to add more emphasizes on edges in addition to what the bilateral filter outputs.

II. METHODS

In our implementation of the sobel filter, one filter is applied to detect edges in the vertical direction and the second filter to detect edges in the horizontal direction (equation 1 & 2). These filters approximate the derivative in their respective directions.

$$Vertical\ Sobel = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Equation (1)

$$Horizontal\ Sobel = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Equation (2)

Figure 1

For each pixel in the frame both these filters are applied to a 3-by-3 matrix of neighboring pixels and summed(equation 3 & 4). The intensity magnitude of the combined outputs (equation 5) produced by these two filters creates a black and white image, where white indicates an edge. A visual example of this filter is shown in Figure 1.

$$V_{m,n} = \sum_{i=m-1,j=n-1}^{m+1,n+1} I_{i,j} * VertSobel_{i,j}$$

Equation (3)

$$H_{m,n} = \sum_{i=m-1,j=n-1}^{m+1,n+1} I_{i,j} * horzSobel_{i,j}$$

Equation (4)

$$I_{m,n} = \sqrt{(V_{mn})^2 + \left(H_{m,n}\right)^2}$$

Equation (5)

The bilateral filter is based on a gaussian filter. When a gaussian filter is applied the image is blurred including edges. This is not a desired effect for rotoscoping. Bilateral filter provides a solution for this problem. The difference between a gaussian filter and a bilateral filter is the way weights are created. For bilateral filter neighboring pixels with similar intensity values as the pixel in question have high weights. Pixels that differ in intensities a lot from the pixel in question have small weights (equation 6). This feature allows for the bilateral filter to preserve edges.

The intensity weights in the previous step are then applied a gaussian filter. Note that the sigma for the intensity weight and the sigma for the gaussian filter are different. This produces the bilateral filter that will be applied to a 3-by-3 matrix to calculate the intensity of the pixel in question (Equation 7).

The recorded intensity for the pixel in question is the summation of the neighboring pixels after applying the bilateral filter (Equation 8).

$$intensityFilter = \sum_{i=m-1,j=n-1}^{m+1,n+1} \left(\frac{1}{2\sigma^2}\right) e^{-\left(I_{i,j} - I_{m,n}\right)^2}$$

Equation (6)

$$bilateral\ filter = gaussian * intensityFilter$$

Equation (7)

$$Intensity_{m,n} = \sum_{i=m-1,j=n-1}^{m+1,n+1} I_{i,j} * bilateralFilter$$

Equation (8)

The final step to produce the rotoscoped image is to subtract the output of the sobel filter from the bilateral output. This essentially darkens the edges making it looked traced as if it was an outline for a drawing.

Figure 2

Figure 3

III. RESULTS

After trying different values for the sigmas, the values that seem to work best were 3 for the distance and 10 for the intensity. Figures 2,3, & 4 show an example frame from a video that the algorithm was applied to. It is clear that it has a cartoony look. Due to a limitation in MATLAB and specifications of the project, only black and white GIFs could be output.

Figure 4

IV. DISCUSSION

As shown by the figures, the output seem as expected, although only with black and white GIFs. In terms of time complexity the algorithm is very slow as it has to go through one pixel at a time on multiple frames. This can quickly lead to slow performance with high quality videos in HD formats or higher. It is also important to note that the output file size is significantly larger than the input file size. This may be due to compression when writing a GIF in MATLAB.

V. CONCLUSION

Applying this filter to different videos revealed the ideal sigmas values of bilateral filter vary depending on the content of the video being rotoscoped. The animations created with this rotoscoping technique are very cartoon looking.

REFERENCES

[1] Edge Detection Sobel Presentation, The University of Auckland, New Zealand
[online] Available at: https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Edge%20detection-Sobel_2up.pdf
[Accessed 6 June 2018].

[2] Bilateral Filtering: Theory and Applications by Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Fredo Durand
[online] Available at: http://www.cse.iitd.ernet.in/~pkalra/col783/bilateral-filtering.pdf
[Accessed 6 June 2018].

Juan Reyes

- Debugged code
- Implemented filters

Marco Chavez

- Debugged code
- Researched digital rotoscoping methods