Automatic Rotoscoping

ECE 160 Multimedia Systems

Antonio Javier Samaniego Jurado
samaniegojurado@umail.ucsb.edu

Pablo Martin Garcia
martingarcia@umail.ucsb.edu

***Abstract*— We propose a video processing algorithm that given an input video creates an automatic rotoscoping effect on it, using intensity adjustment, sharpening, edge extraction, binarization, morphological expansion, and combining frames, and saves it as new output video. We present the methodology, results and discussion in this report.**

I. Introduction

Automatic rotoscoping is an animation technique that is used to convert live-action movie images into animation. Invented in 1920, this technique can be achieved many ways, but in this Matlab© implementation we show how using intensity adjustment, sharpening, edge extraction, binarization, morphological expansion, and combining frames, the result is pretty robust.

II. Methodology

A. Overall Algorithm

```
Input: .filename (video)
Output: rotoscoped, output video
Create video object from filename
input_video ← VideoReader(filename)
Extract input video properties
width ← input_video.width
height ← input_video.height
num_frames ← (input_video.fps) *
             (input_video.duration)

Create output video file
output_video ← VideoWriter(title)
outputVideo.FrameRate ← 10;

for every two frames do
    Sharpening
    Global Intensity adjustment
    Binarization (1st)
    Morphological Expansion (1st)
    Local Intensity adjustment

    if mod(u,5)==0   %Do that every 5 frames
        Add/combine present and previous frame
        Local Intensity adjustment
    elseif mod(u,3)==0   %Do that every 3 frames
        Global Intensity adjustment
        Binarization (2nd)
        Morphological Expansion (2nd)
        Local Intensity adjustment
    previous ← processed_frame
    writeVideo(outputVideo,processed_frame)
end
Plot illustrative results for a single frame
```

Alg 1. Overall Algorithm

First, the algorithm reads an input filename of a video, creating a video objet using VideoReader(). Once the object is created, the algorithm extracts strategic input video properties such as the frame width/height, and calculates the number of frames by multiplying the fps rate by the total video duration.

Next, it creates an output video file by using VideoWriter(), opens the file and sets the output fps rate to 10. At this point, the algorithm is ready to start processing frames for the final output video. It performs all operations every two frames of the input video, so that the final result looks more like a cartoonish animation.

B. Sharpening and Global Intensity Adjustment

First, the algorithm sharpens the current frame that is being processed, using imsharpen().

Then, it adjust the global (all pixels) intensity of the frame using imadjust(). This function maps pixel values from 0 to 1 to a new desired interval between 0 and 1. For esthetic purposes, we have found that mapping to [0.85   1] brightens the frame significantly, since later in the process the object edges areas will become darker to be highlighted. This helps create a pretty strong animation, cartoonish effect.

C. Edge Extraction - First Binarization and Morphological Expansion

A pretty effective technique to achieve the rotoscoping effect is edge detection/extraction.

In this case, first we have performed a binarization of the frame by using imbinarize(), to then apply a morphological expansion to the binarized version using bwmorph(). This last function can either return object edges or their “skeleton”, given a binarized input image. For esthetic purposes, the algorithm performs both, at different times along the ‘for’ loop, as shown in *Overall Algorithm* section, but in this case it first detects the edges by entering argument ‘remove’ to the function. This results in a completely black image (value=0) that contains the image object edges (e.g. people, a car…) highlighted in white (value=1). We later show the result both operations give in further detail (see RESULTS section).

Once that is done, the algorithm, which has the previous step sharpened, brightened version of the original frame, uses the morphologically expanded frame so that the final frame is equal to the brightened version but, wherever there is an edge pixel in the morphologically expanded frame, the value from that same location in the brightened version will be updated using the original frame values (which are way darker) and slightly brightening them, as shown in Alg 2.

```
binarized=imbinarize(frame);
morpho_expanded=bwmorph(binarized,'remove');

for row=1:height
    for col=1:width
        if morpho_expanded(row,col)==1
          frame(row,col)=original_img(row,col)+10;
        end
    end
end

out_frame=frame;
```

Alg 2. Binarization, Morphological expansion and pixel value update.

We have referred to the pixel value update as 'local intensity' adjustment, since it locally changes the dynamic range values of certain pixels.

D. *Frame Combination and Second Binarization and Morphological Expansion - Final Steps*

At this point, the algorithm achieves a pretty good result in terms of how a rotoscoping animation looks like, based on real examples. However, I have gone further and added two more steps. The first is only applied every 5 frames and the second every 3 frames.

The first consists in combining the present frame that is being processed with a stored copy of the previous, already processed frame. Specifically, it adds both frames up (i.e. adds up their pixel values) and divides the result by 2 in order to avoid brightening the image too much. Even so, in order to fix possible 'over-intense' or too dark values as a result of such operation, it performs another local intensity adjustment afterwards.

In particular, given a present frame (the one that is being processed), the previous, already processed frame, and the sum of both, all pixels whose value after the sum operation is below a certain threshold (in this case threshold=140) are updated so that they are assigned that same pixel location value from the present frame. This way, those 'over-intense' pixels recover their original look while preserving the desired 'trail' effect of adding both frames. The process is shown in Alg 3.

```
present_img=frame;
previous_modified_img=previous;

sum_img=(previous_modified_img+present_img)/2;

%Set threshold to perform intensity adjustment:
threshold=140;

%Local intensity adjustment:
for row=1:height
    for col=1:width
        if sum_img(row,col)<threshold
            sum_img(row,col)=present_img(row,col);
        end
    end
end

out_frame=sum_img;
```

Alg 3. Frame Combination (Sum Operation) and Local Intensity Adjustment.

The second and final step consists in analogous to the one described in section *C. Edge Extraction - First Binarization and Morphological Expansion,* but this time, for the morphological expansion operation, it returns the 'skeleton' of the image objects, which even though is not strictly necessary, adds a very particular, animation effect that helps achieve the final rotoscoping effect.

Also, prior to applying such last operations, it changes the global intensity once again in order to give the final video a 'lightning' effect, which also helps achieving the desired animated result.

At this point, the algorithm has completely processed the frame. Next, it writes it into the previously created output video file. Once all frames are written, it closes the video file, saves it to the current directory and the previously indicated title, and plays it.

III. RESULTS

A. *Sharpening and Global Intensity Adjustment Results*

Fig 1. Original/Input Frame

Fig 2. After Sharpening

After choosing a random frame of a sample grayscale video shown in Fig. 1, the resulting frame after applying sharpening can be seen in Fig. 2.

We can see how the sharpening effect is successfully achieved.

After applying the first global intensity adjustment, the result looks as shown in Fig. 3.

Fig 3. After Global Intensity Adjustment

Once again, this change in the global frame intensity is desired to let the edges highlight more later in the process and final result.

B. Edge Extraction - First Binarization and Morphological Expansion Results

The result after applying binarization to the previous step result can be seen in Fig. 4.

Fig 4. After Binarization

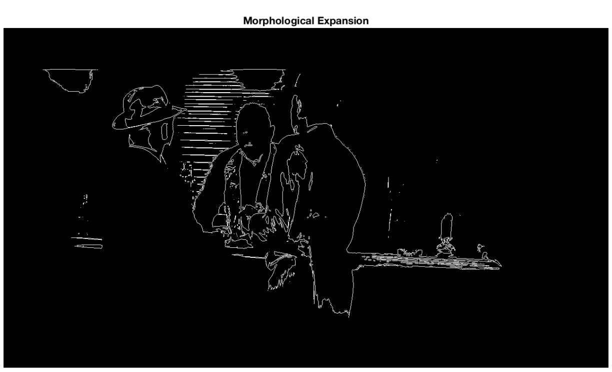

Fig 5. After Morphological Expansion

We can see how based on the original pixel values, they are now mapped to either 0s or 1s depending on their intensity and, after performing the morphological expansion operation to the binarized frame, the edges are pretty decently detected, as shown in Fig. 5, which is a key step in the process.

After both operations and the final pixel value update (local intensity adjustment of the current frame, i.e. the one in Fig. 3, as described in METHODOLOGY section) based on both operations, the result is shown in Fig. 6.

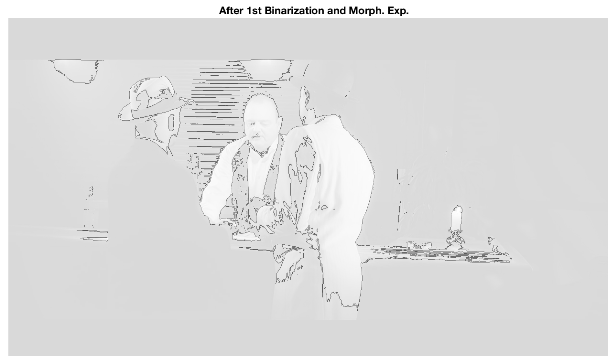

Fig 6. After 1st Binarization, Morphological Expansion and Local Intensity Adjustment

We can see how, at this point, the animated, cartoonish, rotoscoping effect is already achieved pretty robustly.

C. Frame Combination and Second Binarization and Morphological Expansion - Final Steps Results

The added effect of the second binarization and morphological expansion, this time using the ‘skeleton’ argument, can closely be seen in Fig. 7, in form of gentle white lines.

Fig 7. After 2nd Binarization, Morphological Expansion and Local Intensity Adjustment

That along with the frames combination (sum of frames) and the little flashing effect, both previously mentioned and explained in detail, gives the final output video a very strategic trail, animated, effect, which can be visibly appreciated, as the final step of the process. All together achieves a very fair, robust rotoscoping animation, mostly operating with grayscale videos.

IV. CONCLUSION

Automatic Rotoscoping is a very used animation technique that, using the right methods, can be successfully achieved. In this implementation, we have presented how given an input video, performing sharpening, edge detection along with binarization and morphological expansion, strategically combined with changes in global and local intensity, as well as combining frames, results in a very effective way to accomplish this goal.