Modular digital image processing and restoration

Team member:
Wenchen Chen
Jing Wang
Keda Li

Contents

Introduction

The problem faced by image processing is that there are not many user interaction interfaces. The significance of this project is to reduce the difficulty of processing some image problems through the use of GUI (graphical user interface) and VapourSynth based Image processing. The significance of this project lies in the variety of image processing options.

In the GUI, users can select the type of image processing they need and then apply them directly. In the past image processing problems, users often need to modify a large number of parameters and have a certain understanding of programming languages. The purpose of this project is to reduce this gap as much as possible. At the same time, when users have their own personalized needs, through the program commentary. Users can quickly find the parameters that need to be modified, which also ensures the flexibility of the program to a certain extent. The GUI includes the following main functions:1. Get foreground video. 2. Foreground video insertion with mouse click. 3. Faces and other body parts are detected in real time and framed by squares.4. Video tone conversion. 5. video processing with gestures. In the process of two videos combination, there are main video and auxiliary video. The main video size is the original size, and the auxiliary video size is half of the original size. The auxiliary video position will move with the movement of the finger, and the auxiliary video will be split according to the gesture (horizontal and vertical directions).6. Face replacement in a Video7. Video brush. Users can turn on their camera and draw images in the camera based on their gestures. In the following articles, the implementation of specific functions and the applied algorithms will be introduced.8. cartoon processing with different approach.9.graffiti processing, users can change the graffiti position by entering parameters.

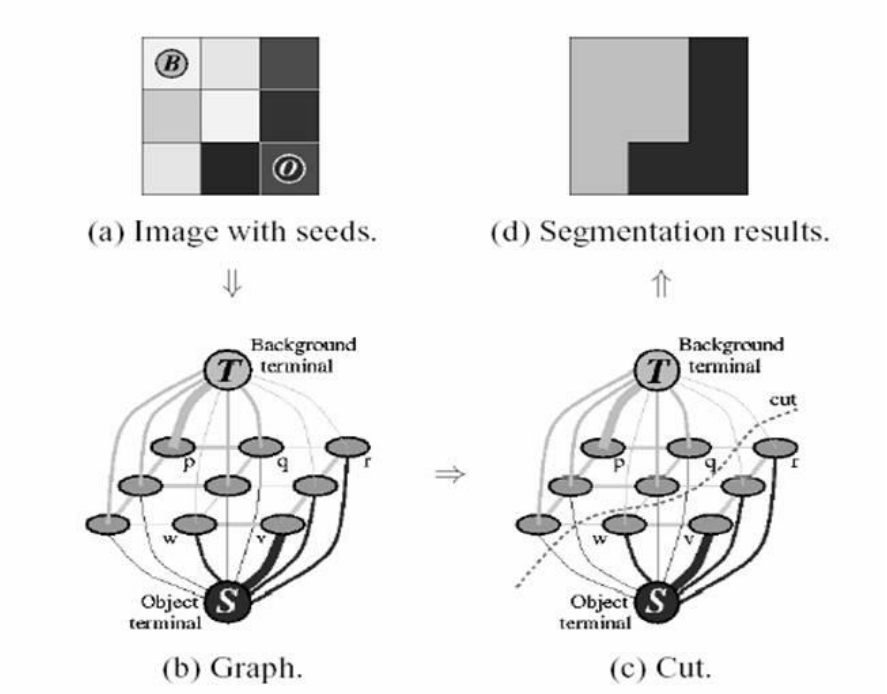
The last part of the project is about high-quality image processing with mask based on VapourSynth, including high-precision image restoration and high-quality upscaling, mainly for 2D-Animation videos. The modules are based on python.

GUI

1. Get the forgroundvideo.

The basic principle is to process each frame of the video with method “applyGrabCut”.
applyGrabCut introduction:

Each pixel will be connected to two nodes(background terminal and object terminal). The goal is to label each pixel as one of these two nodes. The energy function is the key to execution. The energy function consists of two parts, the first part is the weight between pixels, and the second is the weight between pixels and nodes. a large difference in pixel colour results in a very small weight between two pixel. The region information determines the weights between pixels and the two nodes. In order for foreground and background regions to be created, some pixels in the image need to be labelled before segmentation as either foreground or background then A Min-cut/Max-Flow algorithm is used to segment the graph.



When this method is executed, a video named “material.mp4” can be obtained. This video is used as a foreground.

2. Foreground video insertion with mouse click

   The key method here used is setmaterbg(),this method consists of following steps:1.read every frame of foreground and background video.2. resiz the frame of foreground video. This size should at least cover the foreground object.3.convert the frame to gray and create a mask for the foreground image.4.wait for the mouse click event, get the “posx” and “posy”, These two parameters are used to define the position of the upper left corner of the foreground.5.combine fore- and background video. The combination process is as follows. Since the background in the material video is all black, a mask can

be obtained through the threshold method of opencv, and then the bitwesieand function of opencv is applied in the frame where the target object is material to obtain the target foreground. When using “not” on the mask to get the background mask, use the mask and bitwesiand on the background video to get the target background. Finally, the target foreground and target background are combined to get the result

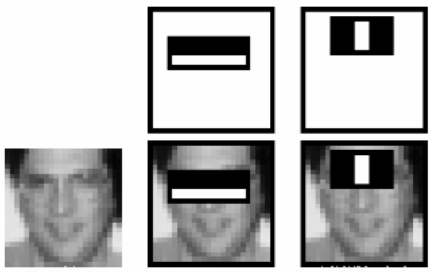
Boundary: When the inserted image exceeds the background range, the excess part will appear on the other side of the background

Situation Analysis:

The blue part is the background plate, the red, yellow and green are the foreground. arrow is the direction of movement. When the foreground movement direction is from top to bottom, the foreground will be divided into upper and lower parts, and the lower part of the foreground will start to appear at the upper edge of the background. When the foreground movement direction is from left to right, the foreground will be divided into left and right parts, and the right part of the foreground will start to appear at the left edge of the background. When the foreground moves diagonally downward, the lower right part of the foreground starts to appear in the upper left corner of the background.

3. Faces and other body parts are detected in real time and framed by squares.

Haar feature-based cascade classifiers is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. The algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then features should be extract from it. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle. Each feature will be used to classify the image. The features with minimum error rate will be selected.



Method introduction: facedet();

The basic idea is to read each frame and then use a detector to detect objects in it.

4. video color conversion

Method introduction: realtimecolortransfer()

The basic idea is to read each frame and then use colorTransfer() form the OpencvTool in the lecture to convert it.

$$I_k = \frac{\sigma_t^k}{\sigma_s^k}(S^k - mean(S^k)) + mean(T^k) \ , k = (\ell, \alpha, \beta)$$

σ_t^k: std of color space in source image

σ_s^k: std of color space in source image

$mean(S^k)$:the mean of color space in source image

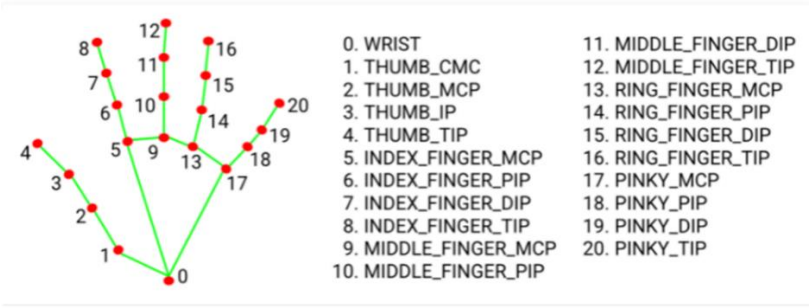
$mean(T^k)$: the mean of color space in target image

$k = (l, \alpha, \beta)$: ℓαβ color space

5. video processing with gestures.

The basic idea is to detect the finger position, then position the target video according to the finger position, and finally put the specified video as the foreground into the background. Gestures can be defined by adjusting the finger distance (horizontal and vertical), and video can be divided horizontally and vertically according to the gesture.

MediaPipe Hands is a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame.



Method introduction:

findpos: this method returns 21 points of one hand. The specified palm position can be obtained by specifying the index.

Gesturedet: This method consists of the following steps:

1). Find all fingertip locations and store them into a list.

2). When the horizontal distance between the index finger and the middle finger is less than 15 pixels, the video is divided horizontally. When the vertical distance between the index finger and the middle finger is less than 15 pixels, the video is divided vertically.

6. Face replacement in a Video.

Basic working principle: first use CascadeClassifier from Opencv to detect the face position and then replace the image at this position with the specified image.

7.Video painter

The basic working principle is to use the forefinger function in Handdectector to detect the position of the index finger, and then use the method of drawing a straight line to connect these points. With the movement of the finger, the thickness and color of the straight line will change within a certain range. Color and thickness variation realization process: Record the number of coordinate points" i "of the index finger on the screen. Divide" i "by 8 and plus 1 to get the remainder, use the remainder as the thickness. Similarly, add "I" to a certain value and divide it by 255 to obtain the remainder. This remainder is used as a color component, and three color components can form a complete color.

8.Cartonnfiy

In order to be able to cartoonise the image, the received image first needs to be converted into a greyscale image, this is where opencv's cvtcolor function is used, then opencv's medianBlur function needs to be used, median blur is achieved by calculating the median of the pixel values that overlap with the kernel and then replacing the central pixel value with the calculated median, next the detection of the The edges of the image, for which adaptive thresholding is applied via opencv's adaptiveThreshold function. The main function of adaptive thresholding is to convert each pixel value in each region of the image to either all-black or all-white based on the average of the pixels overlapping the kernel, and finally using the result of adaptive thresholding as a muster, and the detail based on the value of the muster The enhanced results are merged to obtain a sharp effect with clear edges

9. Graffiti

In order to be able to superimpose a jpg image onto another background, this can be done with opencv.

The first step is to process the received image, for different images such as jpg images and png images, jpg is a BGR three channel image, each channel represents a description of a colour, but png images are four channels, in addition to the BGR channel there is an Alpha channel, which describes the degree of transparency of the image, and the image read with opencv will ignore the Alpha channel, first read the image and then judge the image, whether it is a four channel, otherwise add the alpha channel to the jpg image.

When overlaying images, the png image boundary may exceed the background image due to the inappropriate position of the overlay, for this reason, the overlay position needs to be restricted to meet the image beyond the boundary can still be overlaid normally. Finally, the alpha value of the image to be overlaid is obtained and each channel is overlaid

Image processing based on VapourSynth

To describe the necessity and advantage of our image processing methods, we would like to firstly classify the features of 2D-Animation, of which the most obvious and important thing is that 2D-Animation consist of lines and color block, also called edge area and plane area. Lines and planes of 2D-Animation are very different in nature and require different processing. And there are some common image artifacts of 2D-Animation, such as color banding, aliasing, blurring, oversharp, halos and ringing and the fixing methods to them are various. Unfortunately, most of the time one effective fixing method for certain artifacts can lead to serious side effects in other areas. But with precise mask protecting and 16-bit processing in YUV color space, we can accurately fix the artifacts without bringing side effects. That is why we regard masked-processing as the keypoint of high-quality image processing.

Image processing based on VapourSynth consists of 8 sub-modules, which is:

Module 1 High quality image upscaling method ,

Module 2 denoise with BM3D,

Module 3 various adaptive mask building,

- Module 3.1 luma- and sharpness-adaptive sharpening mask (eemask)
- Module 3.2 luma- and sharpness-adaptive debanding mask (dbmask)
- Module 3.3 luma- and sharpness-adaptive anti-alisasing mask (aamask)
- Module 3.4 credit protecting mask (textmask)

Module 4 masked de-band with f3kdb (deband)

Module 5 masked anti-aliasing algorithm with multiple kernels (anti-aliasing)

Module 6 luma- and sharpness-adptive UnsharpMask sharpen (sharpen)

Module 7 Super HD de-ringing and de-haloing (dering)

Module 8 masked luma-adptive denoise. (masked denoise)

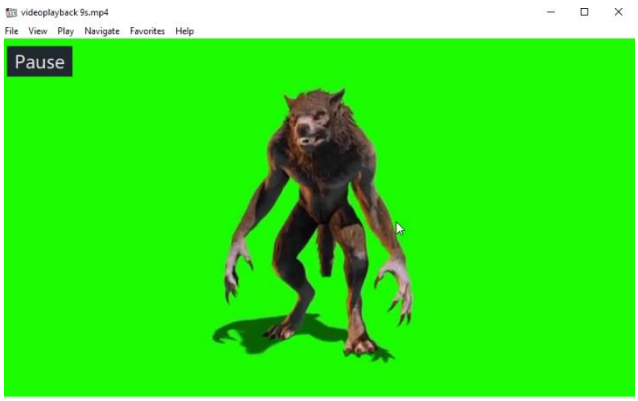
The code that implements the method has been commented in detail in DBV image restoration main.vpy script. The structure and order of implementation of the scripts are exactly the same as those listed above. With the comments, you can easily understand the implementation of all methods. To go further into kernels of our processing methods, please read the main.vpy script.

We first briefly introduce the extra artifacts that processing methods above could bring in, and how we can counteract side effects by arrange them together. All steps are performed in 16-bit YUV colorspace. First of all we always use Denoise algorithm like BM3D as low-pass filter to separate noise layer. Denoise and Anti-aliasing processing could destroy the texture and blur lines, while de-banding could smooth or even eliminate small and faint lines , especially in dark area. In order to make up for it, we need masked adaptive-sharpening to enhance the texture and sharpen most lines, excluding strong lines in bright area. But even if the sharpening strength is controlled by the eemask, it is difficult to avoid ringing and haloing caused by excessive sharpening. So next we must apply de-ringing method. In order to protect credits from de-ringing procedure, textmask is necessary. Then we overlay the noise reduced by noisemask onto the image. At last if we plan to encode the video, we can just set the output clip to 10-bit or 8-bit and transfer it to x265 or x264 video encoder.

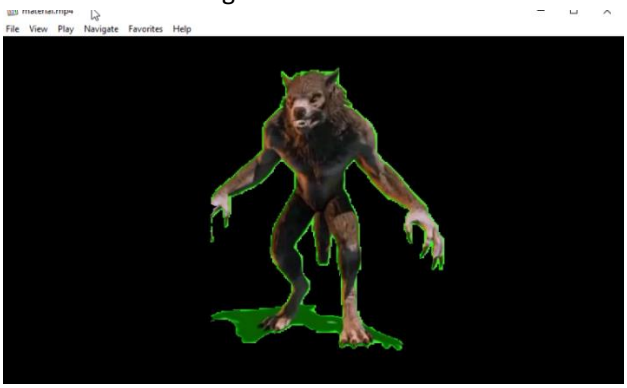
Results presentation and evaluation

GUI

1. Getforgroundvideo



Original video



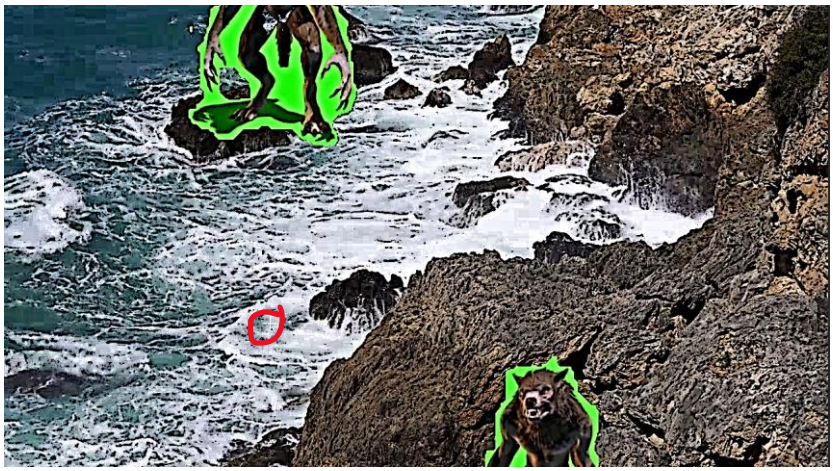
Video after processing

```
GrabCut took 1.57 seconds.
GrabCut took 1.51 seconds.
GrabCut took 1.43 seconds.
GrabCut took 1.47 seconds.
```

The time required to process each frame of the picture

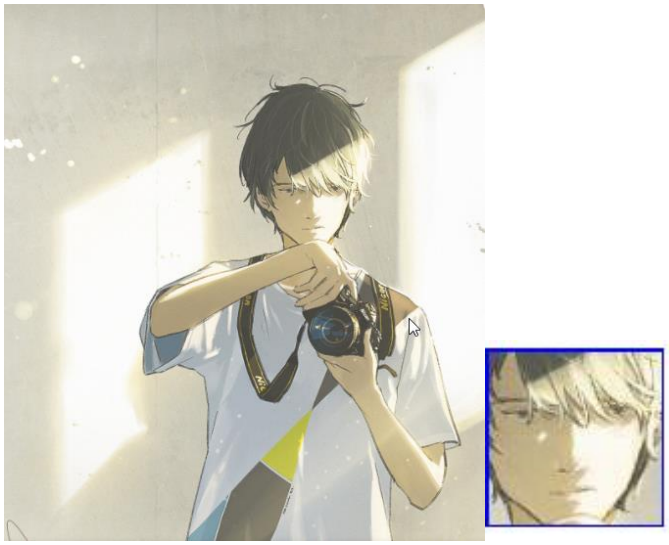
The interception effect meets the expected requirements, but some background will be intercepted and the processing time is long. When the video time is long or the video resolution is high, the time consumption will increase quickly.

2. Foreground video insertion with mouse click



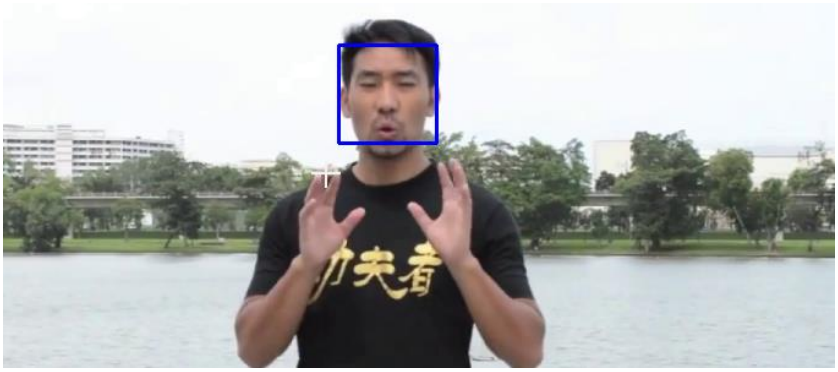
The red position is the mouse. The lower body of the wolf will appear from the top edge of the background. The effect meets the expected requirements, but there may be some lags during processing

3. Cropandsavepic



The left is the actual picture, the right is the extraction effect.When the facial features are obvious, the effect of taking the image is good(original pic(left),cropped pic(right).

4. Realtimedet



Faces can be accurately identified

5. Colortransfer

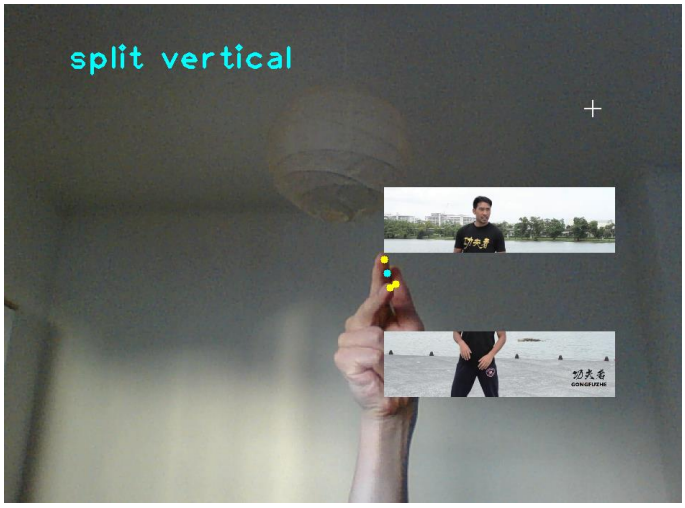
original video

tone picture

processed video

The top image is the original image, the middle image is the tone image, and the bottom image is the result. The color conversion effect is obvious

6. Videocontrolwithgesture

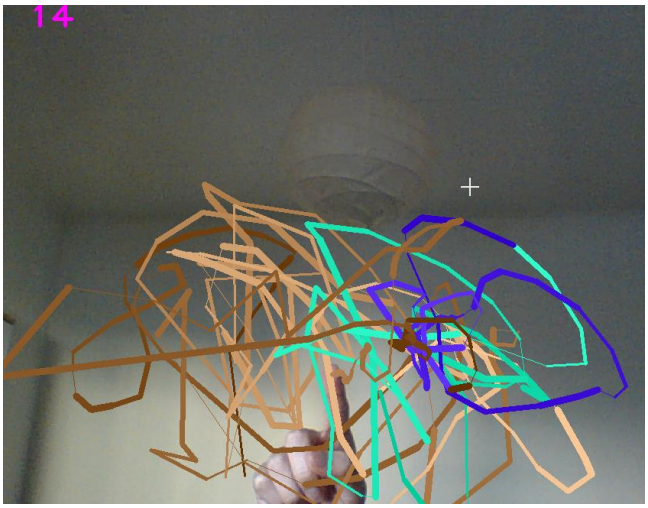


When the horizontal distance between the index finger and the middle finger is less than 15 pixels, the foreground video will be divided into left and right parts. When the vertical distance between the index finger and the middle finger is less than 15 pixels, the foreground video will be divided into two parts. upper and lower parts. The required functionality was successfully implemented

7. Facereplace

Human face is successfully replaced.

8. Videopainter



The width and color of the line changes as the finger moves. The pink number in the upper left corner is FPS (frames per second). The track of the index finger is tracked correctly.

cartoon effect

graffiti effect

The results are shown in above and are tested on different image formats and the runtime for cartoonisation and overlay is very short and almost ignorable, but graffiti has obvious limitations in that it can only overlay images, so if you need to overlay a video, you need to do additional processing on the video, and in addition, graffiti does not handle the details of some images very well.

Cartoonisation and graffiti are able to process the input material correctly and output the results, but there are limitations in terms of extending the extra functionality, for which some improvements are needed, such as more detailed processing.

VapourSynth

Since various masks are so important in the project, we should introduce how mask works in VapourSynth. Masks can perform precise pixel manipulations on image areas of different features like a scalpel. MaskedMerge merges clipa with clipb using the per pixel weights in the mask, where 0 means that clipa is returned unchanged. The picture below can help you understand.

And there is another example below. The left original image of left girl is clipb, and the right one is clipa, and the black and white illustration is mask clip.

After that we would also show different masks that we have built in main.vpy script. In mask clip, the bigger Y-value pixel has, the brighter area of mask clip is. lease open the folder “mask sets” and view each set of pictures alternately at original scale while reading the following descriptions.

(1) eemask is a continuous mask, mainly used to sharpen the line-area and texture-area. Sharpness changes in low-luminance areas are more obvious for human eyes, so we take both line-strength and luma into consideration when sharpening the image. Sharp area and bright area are of higher pixel value than soft and dark area do. According to the sharpening-strength curve we established, low-brightness areas and less-sharp areas such as textures are significantly sharpened.

(2) dbmask is a binarized mask, which can fix corlor-banding without destroying textures. It covers as many lines and textures as possible to prevent them from being smoothed by de-banding filters while debanding filters fix corlor banding. Meanwhile the edges of color banding should not be recognized as textures and covered by dbmask

(3) aamask is a binarized mask, which can smooth lines, fix aliasing and link broken lines without destroying textures. It only cover strong lines but exclude textures.

(4) credit protecting mask is a binarized mask, which can protect credits, staff list and subtitles from being damaged by other artifacts fixing filters. It covers only credits.

(5) noisemask is a continuous mask, which determines the strength of denoise. Human eye is more sensitive to noise in dark area than that in bright area. So we build a luma-adptive called nrweight to control how much noise should be retained in different brightness areas.

Then we would like to present the results of our image preocessing methods. Please open the folder “comparison sets” and view each set of pictures alternately at original scale, and carefully find the differences with the help of following instructions.

(1) For comparison set 01, please notice the aliasing on the edge of the arm and tablet. There is also a ringing around the aliasing. This set shows the effect of anti-aliasing and de-ringing.

(2) For comparison set 02, please notice the noise and luma-banding and chroma-banding in night sky. They make the image look dirty. This set shows the effect of de-banding and denoise.

(3) For comparison set 03, please notice the edge of the girl and texture area of grass. In output image, they all look sharper and clearer. This set shows the effect of sharpening without de-ringing.

(4) For comparison set 04, please notice the color banding on the girl’s face and hair. We fix it without blurring the lines or smoothing the weak textures like eyelashes and bangs. This set show the effect of de-banding. This set shows the effect of de-banding.

(5) For comparison set 05, please notice all lines. Everything looks clearer in output image than that in the original image. This set shows the effect of high-quality image upscaling and sharpening with de-ringing.

(6) For comparison set 06, please notice the girl’s profile, clothes and hair. Lines in the original picture are broken and blurred and surrounded by slight ringing. Meanwhile, lines in output picture are smooth and darker and the ringing is gone. This set shows the effect of anti-aliasing and sharpening with de-ringing.

(7) For comparison set 07, please notice the left girl’s hair, neckline and the wood texture on

the windowsill. We fixed the strong aliasing and made the lines sharpener without bring any extra ringing. And the wood texture are also clearer. This set shows the effect of anti-aliasing and sharpening with de-ringing.

(8) For comparison set 08, please notice the mosquito noise in night sky. The output picture seems apparently cleaner than original one and no texture gets blurred. It also reduce the file size of image. This set shows the power of denoise.

We achieved very good results by proposing various processing methods with masks. This set of scripts is versatile and can be used for various 2D-animations.

Improvement ideas

Before the getforegroundvideo method is performed, the target object needs to be selected with a rectangular frame, and the source code(rec) needs to be modified to implement this process, and the position of the frame-selected object cannot be moved too much, which is a limitation of the current method. In the future, machine learning methods can be used to track the characteristics of the target object to reduce the difficulty of user operations. There is also a lot of room for improvement in processing speed.

Currently the graffiti method is limited to certain specific image formats and can only process images, it would be better if this function could process images of any format, in addition, graffiti this function can only do the overlay of the image, can not make it have a more three-dimensional effect, there is no deformation of the image to fit the background image.

As for cartoonization, there are many ways to achieve the cartoon effect, but the effect of each way is not the same, if this can combine the advantages in various ways, it will have better result.

For the VapourSynth. As far as I know, there is no publicly available full set of processing scripts in the field of image processing based on VapourSynth. So I didn't find a comparison object. But I think this project still has room for improvement. For example, the high-quality image upscaling kernel of this project still does not deviate from the traditional algorithm. In fact, it is possible to try machine learning to train neural networks for image upscaling, such as waifu2x.

Reference

Reinhard, E., Adhikhmin, M., Gooch, B., & Shirley, P. (2001). Color transfer between images. IEEE Computer Graphics and Applications, 21(4), 34–36. https://doi.org/10.1109/38.946629

https://docs.opencv.org/3.4/d8/d83/tutorial_py_grabcut.html

https://www.youtube.com/watch?v=01sAkU_NvOY&t=11350s

https://google.github.io/mediapipe/solutions/hands.html#model_complexity

https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

Filters Reference:

https://amusementclub.github.io/doc3/includedplugins.html

https://amusementclub.github.io/doc3/functions.html

https://github.com/HomeOfVapourSynthEvolution/mvsfunc

https://github.com/HomeOfVapourSynthEvolution/VapourSynth-BM3D

https://github.com/HomeOfVapourSynthEvolution/havsfunc

https://github.com/HomeOfVapourSynthEvolution/VapourSynth-TCanny

https://github.com/HomeOfVapourSynthEvolution/VapourSynth-EEDI2

https://github.com/sekrit-twc/znedi3

https://github.com/dubhater/vapoursynth-sangnom

https://amusementclub.github.io/doc3/plugins/rgvs.html

https://github.com/SAPikachu/flash3kyuu_deband

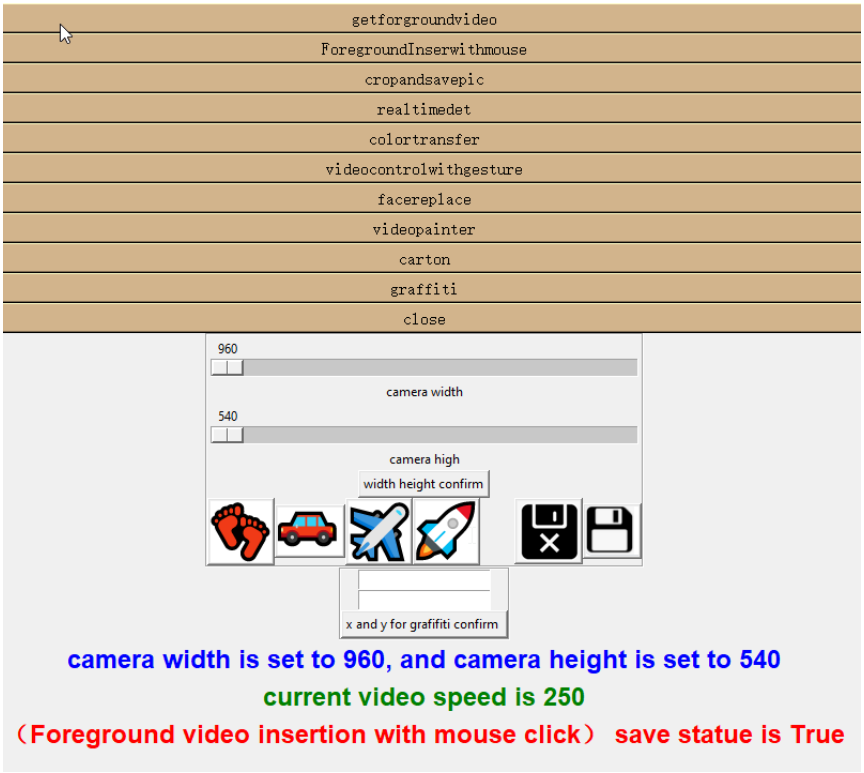
https://github.com/dubhater/vapoursynth-awarpsharp2

https://github.com/EleonoreMizo/fmtconv

http://www.msystem.waw.pl/x265/

Appendix

GUI appearance



Button introduction

The slider above can control the size of the camera

Foot, car, plane, rocket represent the speed gear of the video, the formula is：

$$f = \frac{1000}{s}$$

f: The interval between frames. Unit is in milliseconds

s: speed class. There are four choices:4,40,250,500

save/no save. These two buttons control whether to save the result of Foreground video insertion with mouse click.

x and y for graffiti. This button is used to determine the location of the graffiti in the “graffiti” method

Installation guidance of VapourSynth

(1) Dowmload the portable package of VapourSynth, in which common filters and libs are included. It supports only windows.

(2) Open .vpy script with vsedit.exe and follow the instructions inside the script. F6 is grammar check and F5 is preview.

(3) If everything is ok, set Debug=0 and save the script.

(4) Edit the encode.bat, and set the path to vspipe.exe and x265-10bit_asuna.exe. Save profile.

(5) Then drag the .vpy to encode.batch, or you can use your own style to start video encoding.