

# **Comp Photography Final Project**

Jialin Yu

Fall 2016

[jyu334@gatech.edu](mailto:jyu334@gatech.edu)

# Minion & Tony

One of my 6-year-old son (Tony)'s dreams is celebrating Christmas with his favorite character, minion! So here comes my idea, record a video for minion and Tony before a Christmas tree, isn't it cool? So what I did was record a video with everything except minion, and split them into frames, then add 3D minion to each frame with same camera perspective and lighting situation, and merge all the frames back to movie.

# The Goal of Your Project

A paragraph or two describing the original goal (scope) of your project. What motivated you to do this project?

I myself is very interested in scientific movie creation process, the CGI technique with green screen are very cool and I wanted to do something similar here. My son is a big fan of minion, so I had the idea to bring 3D minion life in my son's video. The scope of the project is somehow beyond cv2, I used Blender as my 3D tool to generate minion frames, there were a lot of steps in setting up scene with same lighting condition in my video, same camera perspective and same pace to move the camera view. After all the frames are generated, I used cv2's alpha blending to add minion as a top layer on the original frame.

# Scope Changes

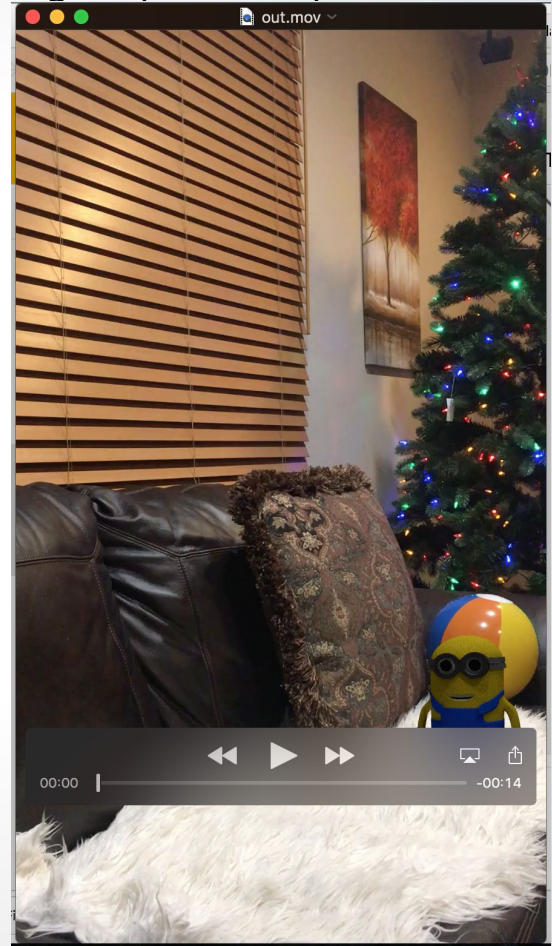
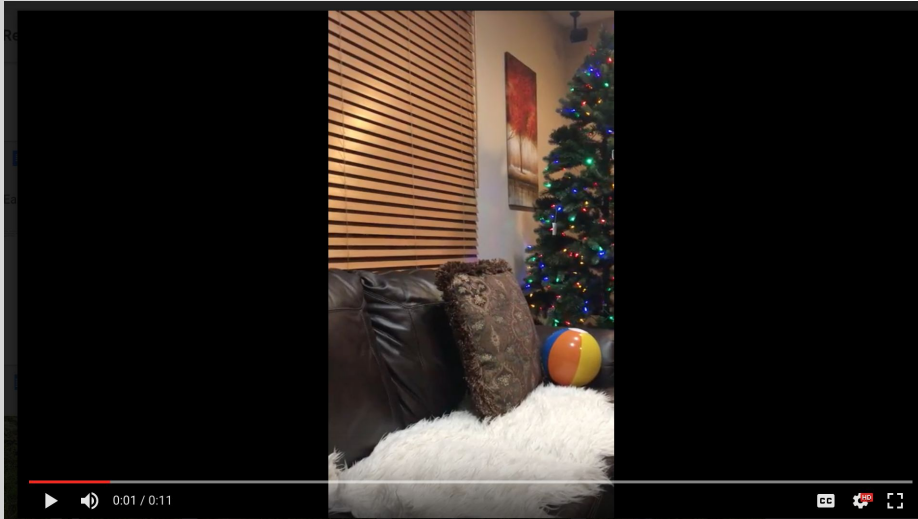
Did you run into issues that required you to change project scope from your proposal?

- If so, give a short explanation here of what changed.
- Go into more depth in a later section.

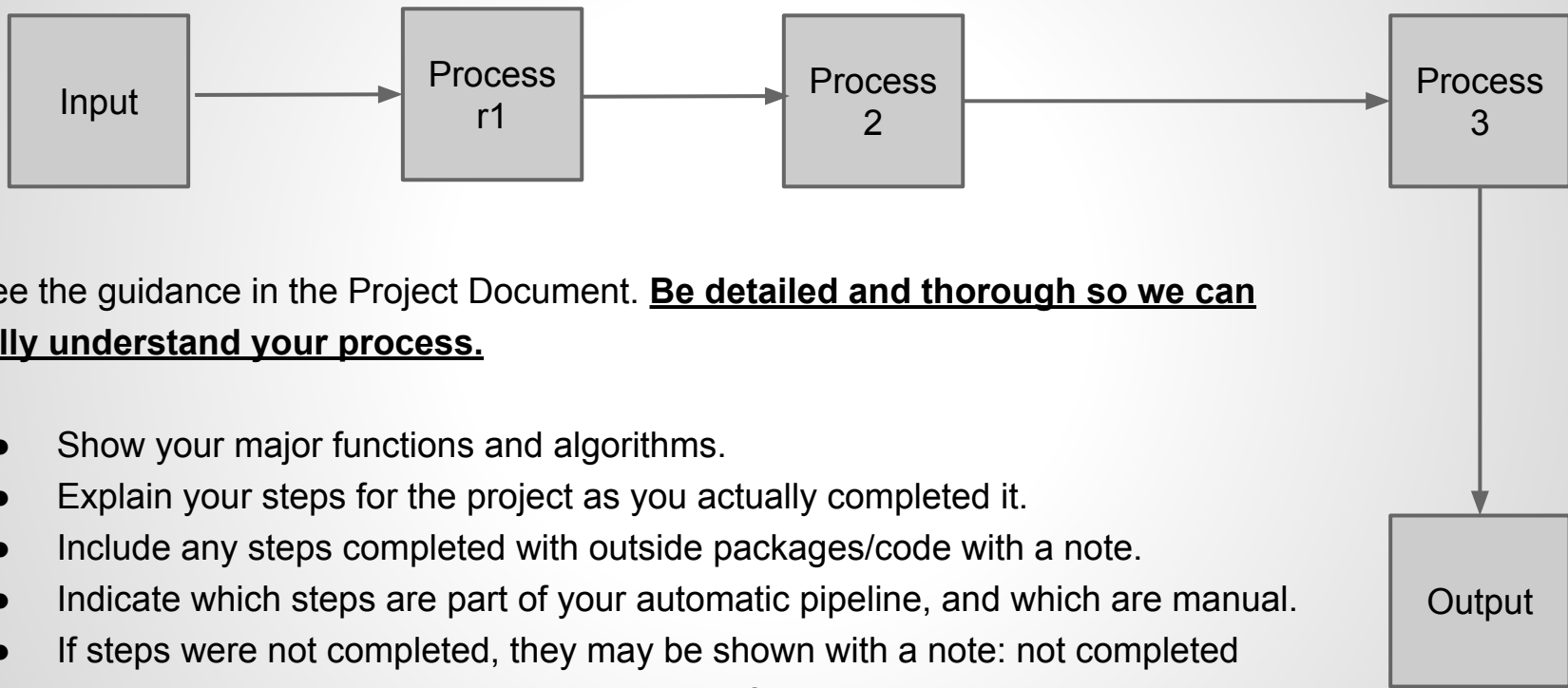
Yes only slightly. I was planning to replace the ball object in original video with the minion object, but with difficulties. So finally I use the ball as a reference only and add the minion object directly on the frames.

Showcase what you did on **This One Single Slide**. That might be challenging. You may use several images and format how you wish; but this single slide should be a good pictorial representation of your work. Be creative.

Input



# Your Pipeline



See the guidance in the Project Document. **Be detailed and thorough so we can fully understand your process.**

- Show your major functions and algorithms.
- Explain your steps for the project as you actually completed it.
- Include any steps completed with outside packages/code with a note.
- Indicate which steps are part of your automatic pipeline, and which are manual.
- If steps were not completed, they may be shown with a note: not completed
- You may split the pipeline to another page, if necessary.

Input: a video for 11s

Process r1: use ffmpeg to split the video into 358 frames with fps 25.

Process r2: Set up the scene for a minion 3D model inside Blender(v2.78), including add 4 recess light on same position in the scene and setup a front camera at same distance And setup the frame timeline as same as the original video. So that the minion object in each frame has best simulated lighting, position and view perspective.

Process r3: running my python script inside Blender to generate image per frame position, with an alpha channel, so that other than the minion object, it is transparent.

Process r4: looping through all the original frame and blender frame pair. Add cropped Minion with alpha channel as a top layer onto original frame.

Process r5: merge all the new frames back to video, using same fps.

Output: a new video with Minion as a 3D object inside.

# Demonstration: Show complete Input/Output results

- Check slide #5 for Input and Output results with links
- Below are intermediate input and output results
- Input frames after splitting original video:  
<https://drive.google.com/open?id=0B3R51Ln0szVmNIZCdTI3SEM0R3M>
- Input frames as blender rendered frames:  
<https://drive.google.com/open?id=0B3R51Ln0szVmNIZCdTI3SEM0R3M> [SSDJCQjVkdFU](https://drive.google.com/open?id=0B3R51Ln0szVmNIZCdTI3SEM0R3M)
- Cropped minion example----->
- Output combined frames:  
<https://drive.google.com/open?id=0B3R51Ln0szVmNIZCdTI3SEM0R3M> [ZpbHBfaXFkUDQ](https://drive.google.com/open?id=0B3R51Ln0szVmNIZCdTI3SEM0R3M)





# Computation: Project Development

Use several slides for a detailed discussion of how you developed your code and project outputs. Tell your story.

This can take many forms:

- Narrative of your progress and issues you faced
- Include descriptions of libraries/programs that you used (for minor sections)
- Code-based explanations
- Include descriptions of problems, and how you handled them. Including poor results that lead to changes is often helpful.
- Use as many pages as necessary.

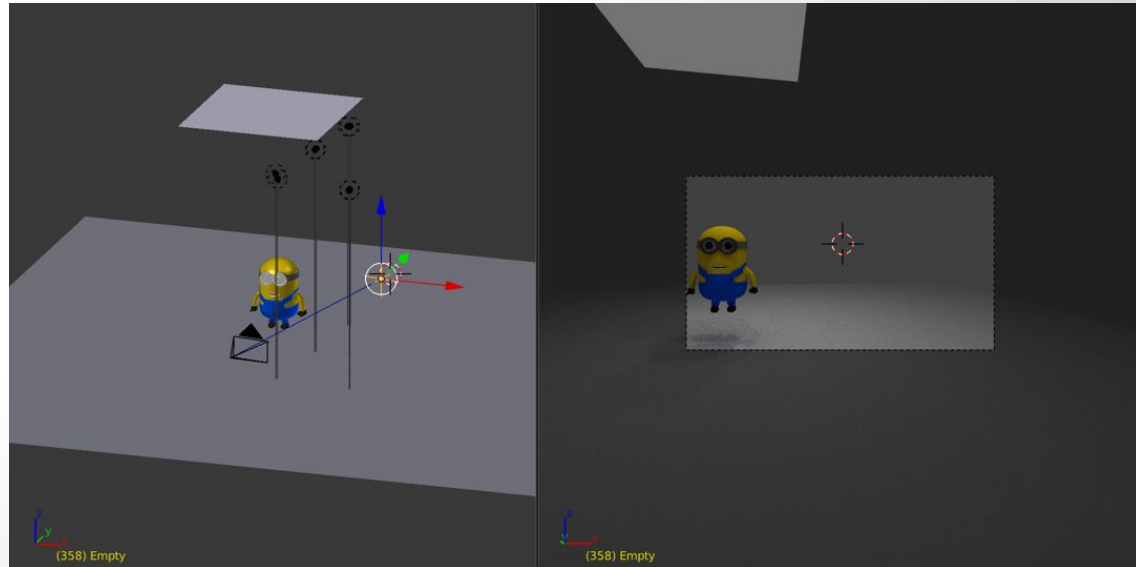
# Process r1

Used ffmpeg to split my 11s video.

```
ffmpeg -i video.ext -r 1 -f image2 image_directory/%04d.png
```

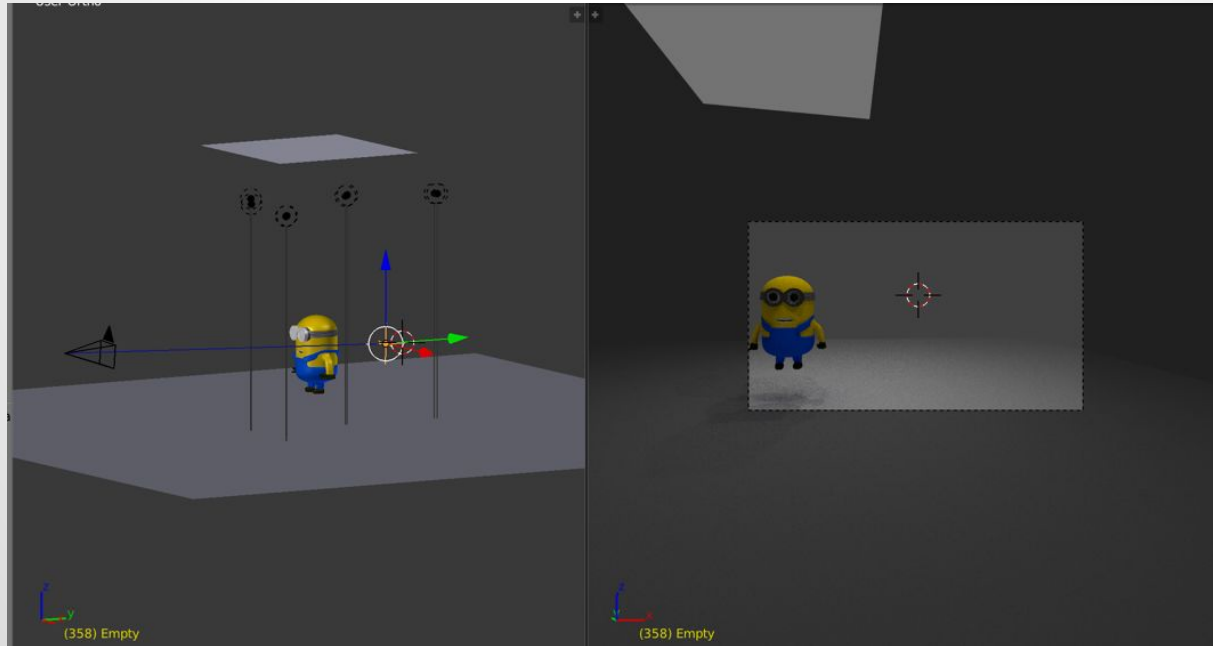
# Process r2

Setup an active scene in Blender and put an minion object downloaded online into the scene. Then setup the lighting, exactly same location as recess lights in my living room.



# Process r2

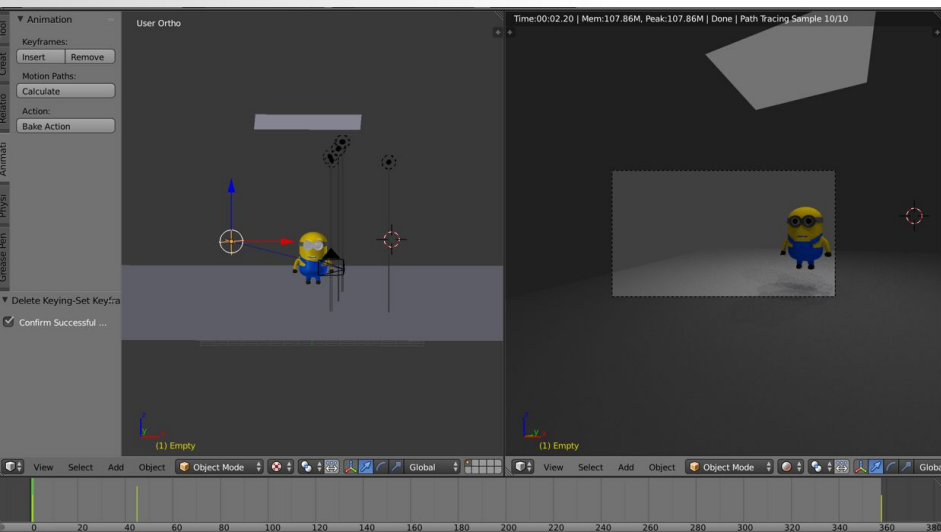
Setup the front camera, roughly same distance and height. Right window is camera perspective view, trying to match my original video. I use a ball in original video as reference.



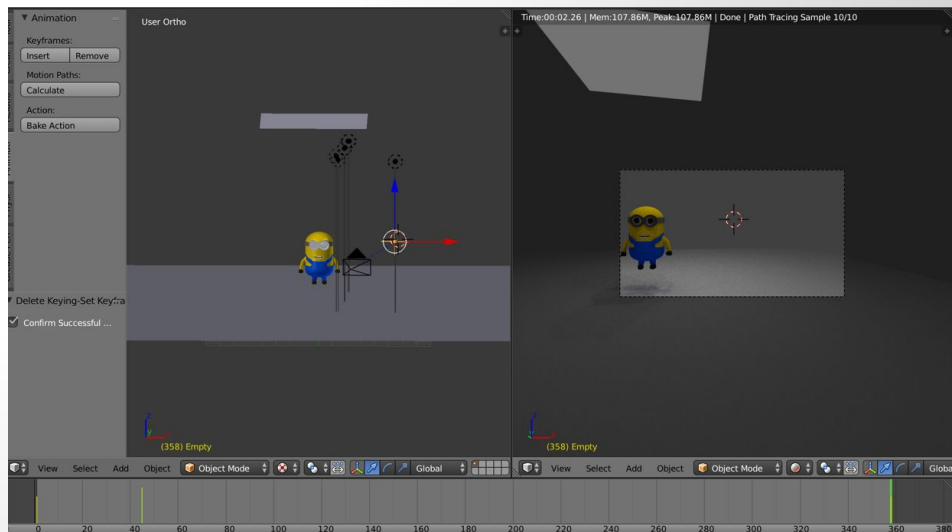
# Process r2

Setup the timeline with same frame at same begin and end position, since my camera swing back and forth once, I added 3 frame position for minion in Blender. Again, the begin and end frame location try to match the reference ball location in original video's start and end.

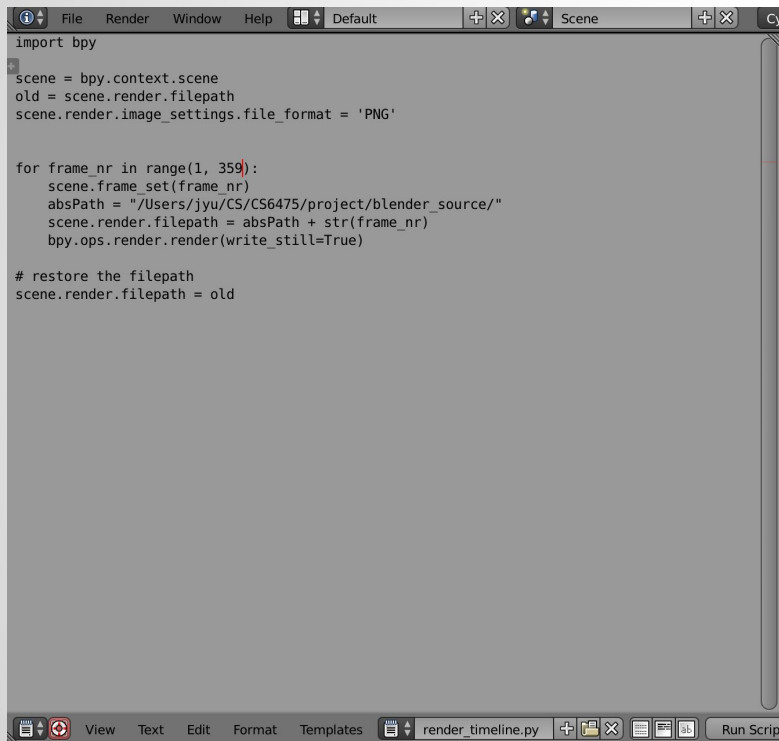
Frame 1



Frame 358



# Process r3

A screenshot of the Blender 2.79 Python console window. The window has a title bar with 'File', 'Render', 'Window', 'Help', and 'Default' menus. The main area displays a Python script for rendering frames. The script imports bpy, sets the scene, and loops through frames 1 to 359, setting the frame set, absolute path, and rendering each frame. It also includes a comment and code to restore the original filepath. The bottom status bar shows 'render\_timeline.py' and a 'Run Script' button.

```
import bpy

scene = bpy.context.scene
old = scene.render.filepath
scene.render.image_settings.file_format = 'PNG'

for frame_nr in range(1, 359):
    scene.frame_set(frame_nr)
    absPath = "/Users/jyu/CS/CS6475/project/blender_source/"
    scene.render.filepath = absPath + str(frame_nr)
    bpy.ops.render.render(write_still=True)

# restore the filepath
scene.render.filepath = old
```

In order to generate 358 frames, I wrote a python script and run it inside Blender, so Blender can render the image based on the timeline I setup with the locations.

So this step mimic the manual recording for minion and then split the video into same number of frames as the original one. Hence the view perspective and shadow condition can be roughly the same in the real scene.

# Process r4

Now I have two sets of frames, one is original frame, and the other is minion frames. Next I need to blend them. Since minion frames are generated purposely with alpha channel to make the background transparent, I don't need to use any mask. So first I try to get a minion center. This can be done by collecting all none-0 channel and get x, y mean of locations. Then I crop the minion into smaller png. The offset is setup manually by trying to see if the whole minion can be cropped.

```
def get_object_center(img):  
    ptx = []  
    pty = []  
    for i in xrange(img.shape[0]):  
        for j in xrange(img.shape[1]):  
            if img[i][j].all() != 0:  
                ptx.append(i)  
                pty.append(j)  
    center = (sum(ptx)/len(ptx), sum(pty)/len(pty))  
    return center
```

```
def crop_object(img, center):  
    x_offset = 190  
    y_offset = 110  
    return img[center[0]-x_offset:center[0]+x_offset,  
               center[1]-y_offset:center[1]+y_offset]
```

# Process r4

Finally I did a alpha blending, adding minion as top layer onto original frame. There are same tricks indeed. Since I only rotating my camera vertically, I assume the x-dimension is fixed. By manually adjust the x insert location, I can fix it in all the frame blending. The y-dimension is a little tricky. Since my Blender generated view is mimicking my original view perspective. I assume they are exactly in the same pace. So I take a relative position of minion's center to calculate the absolute y location of insert point in the original frame.

```
#cv2.imwrite('tmp0.png', top_layer)
relative_y = (center[1]-top_layer.shape[1]/2) *1.0 / black.shape[1]
insert_y = int(relative_y * bottom.shape[1])

def alpha_blending(top, bottom, insert_y):
    fixed_x = 1000
    for c in range(0,3):
        alpha = top[:, :, 3] / 255.0
        insert_x = fixed_x + top.shape[0]/2
        bottom[insert_x:insert_x + top.shape[0], insert_y:insert_y+top.shape[1], c] = \
            top[:, :, c] * alpha + \
            bottom[insert_x:insert_x + top.shape[0], insert_y:insert_y+top.shape[1], c] * (1-alpha)
    return bottom
```



# Process r5

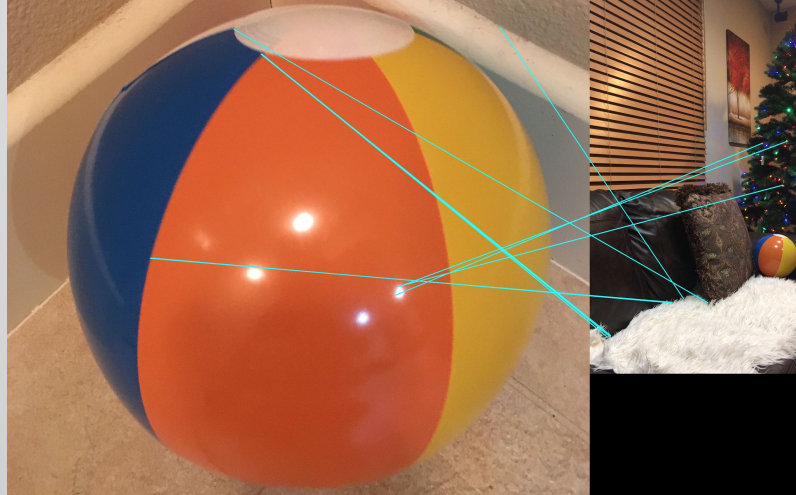
```
ffmpeg -framerate 25 -pattern_type glob -i '*.png' -c:v  
libx264 -pix_fmt yuv420p out.mov
```

In this step, I simply merge the images back and the fps is set to original value which is 25.

# Issues run into

1. Even though the light and shadow effect on the minion object self is good enough, the shadow on the ground cannot be simply picked up, otherwise the ground object needs to be picked up as well. To take advantage of alpha channel, the background transparency is necessary, so the ground object is very tricky to be included. There are suggestions online to do it with node layer addition, but I was not able to reproduce.
2. Since the original video was taken by hand, I cannot control the camera rotation speed is exactly the same all the time, so some of the layer insert position is shifted and the it looks like minion is floating left and right for some time in the final result. I redid the record for several times and still have some speed variation and I assume it cannot be avoided if done manually.

# Issues continued



3. My original plan was to replace the ball on the sofa with minion, and I planned to find out the ball's center fist, so that I could determine the insert position. But it turns out really hard to do so. I try to do feature match with a large scale ball. But the result is very poor. Also, I tried use one of the RGB channel, also did not work, since too many similar color in the image. I also tried findContours in cv2 after I convert it to gray, still too many objects and the contours are connected to each other. So finally I gave up the idea to replace the ball, just add minion on.

# Details: What worked?

- **Detailed discussion!**
- Maybe you already covered this in your narrative, if so, you may delete this page.
- We want to remind you again, just in case.

# Things worked well

I want to add this page to specifically discuss the Blender approach to generate frames. My original plan is to use a real toy minion at my home to do the job, I planned to add markers in the original scene and the real minion scene and then match the markers together when adding minion objects. But this has a lot of difficulties. My background is very colorful, it's hard to add markers in this situation while in movie making, people using green screen or object with markers. Then most importantly, I need to match camera angle in the two scenes, I think I was not able to do it, besides, the lighting condition needs to be the same to reproduce shadow. So I switched to Blender, a really good 3D tool. With the tool, lighting could be simulated to be the same and each frame is precisely controlled by the preset angle, and the best thing is I could render each frame with a transparent background which is amazing for layer blending. I don't need to worry about how to get my object out of the image. And the final image quality is so good.

## Details: What did not work? Why?

- **Detailed discussion!**
- Use additional slides to present failed outputs if needed (recommended)
- If you have covered this very well in your narrative, delete this page.

Refer to my slide #17 and 18

# Computation: Code Functional Description

Walk through your code functions, similar to what was done in assignments.

- Show and discuss code snippets. Explain the purpose of the code you're presenting.
- Explain the major algorithms. If you used them, show that you understand them.
- Code discussion may be incorporated in the Project Development area if that works better for you.
- Make sure you credit your sources explicitly, either here or in Resources.
- Use as many pages as necessary.

Check slide #14-16

# What would you do differently?

1. I would use a different scene with less similar colors, for example a room with light pink color and put a marker objects with dark green. Then this marker object can be detected if I turn only the green channel on. And I can replace the marker with my target object if I get the center of the marker.
2. I would fix my camera, and let the object move, for example, ask my son to grab the marker to play around, so that it can move. Since I can always get marker center from step 1, I can replace it in each frame.
3. Some smart blend methods, since the marker is grabbed by a human, it will be sometimes blocked by human body parts. I will do some if else tricks to only replace the visible part of marker.



# Any additional details?

- Last chance.
- Discuss things that don't seem to belong anywhere else.
- Delete this page if you don't need it.

# Resources

- <http://tf3dm.com/3d-model/minion-73703.html>
- [http://docs.opencv.org/2.4/doc/tutorials/core/adding\\_images/adding\\_images.html](http://docs.opencv.org/2.4/doc/tutorials/core/adding_images/adding_images.html)
- <https://www.youtube.com/watch?v=jJqcpaDTr7s>
- <https://www.youtube.com/watch?v=fgAySB7DoOc&t=693s>
- <https://www.youtube.com/watch?v=appYBmrhWJE>
- <http://blender.stackexchange.com/questions/17839/python-render-specific-frames>
- <https://cgcookie.com/flow/introduction-to-blender/>
- <http://trac.ffmpeg.org/wiki/Create%20a%20video%20slideshow%20from%200images>

# Appendix: Your Code

We do not have a preference for the form of your code submission. We rarely run project code, but we do inspect it.

Copy your code here, use as many pages as you need. Use whole slides without titles.

Exceptions:

- You provide a working link to your code (i.e. github)
- Your code was very extensively described and shown in your functional description

You may use small type (such as this Arial 10) but please, nothing smaller.

main.py

```
import cv2
```

```
def get_object_center(img):  
    ptx = []  
    pty = []  
    for i in xrange(img.shape[0]):  
        for j in xrange(img.shape[1]):  
            if img[i][j].all() != 0:  
                ptx.append(i)  
                pty.append(j)  
    center = (sum(ptx)/len(ptx), sum(pty)/len(pty))  
    return center
```

```
def crop_object(img, center):  
    x_offset = 190  
    y_offset = 110  
    return img[center[0]-x_offset:center[0]+x_offset,  
               center[1]-y_offset:center[1]+y_offset]
```

```
def alpha_blending(top, bottom, insert_y):  
    fixed_x = 1000  
    for c in range(0,3):  
        alpha = top[:, :, 3] / 255.0  
        insert_x = fixed_x + top.shape[0]/2  
        bottom[insert_x:insert_x + top.shape[0],  
insert_y:insert_y+top.shape[1], c] = \  
            top[:, :, c] * alpha + \  
            bottom[insert_x:insert_x + top.shape[0],  
insert_y:insert_y+top.shape[1], c] * (1-alpha)  
    return bottom
```

```
def main():  
    for i in xrange(1, 359):  
        name1 = str(i) + '.png'  
        name2 = '{0:04d}.png'.format(i)  
        black = cv2.imread("blender_source/" + name1, -1)  
        bottom = cv2.imread("source/frame" + name2, -1)  
        center = get_object_center(black)  
        top_layer = crop_object(black, center)  
        #cv2.imwrite('tmp0.png', top_layer)  
        relative_y = (center[1]-top_layer.shape[1]/2) *1.0 / black.shape[1]  
        insert_y = int(relative_y * bottom.shape[1])  
        result = alpha_blending(top_layer, bottom, insert_y)  
        cv2.imwrite('output/frame{0:04d}.png'.format(i), result)
```

```
if __name__ == "__main__":  
    main()
```

## render\_timeline.py

```
import bpy
```

```
scene = bpy.context.scene
```

```
old = scene.render.filepath
```

```
scene.render.image_settings.file_format = 'PNG'
```

```
for frame_nr in range(1, 359):
```

```
    scene.frame_set(frame_nr)
```

```
    absPath = "/Users/jyu/CS/CS6475/project/blender_source/"
```

```
    scene.render.filepath = absPath + str(frame_nr)
```

```
    scene.render.alpha_mode = 'TRANSPARENT'
```

```
    bpy.ops.render.render(write_still=True)
```

```
# restore the filepath
```

```
scene.render.filepath = old
```

# Credits or Thanks

- Thanks to the professor and TAs who worked diligently in this semester and thanks a lot to the wonderful CS 6475!!!
- Thanks to my husband who coaching my son's piano during the time I was working on the project.
- Thanks to my son who patiently waiting for me to finish the project and his nice performance in the video.
-