Computational Photography
Assignment #7
Video Textures

Maksim Kazakov
Fall 2018

Best Results from Sample Input

*Start frame filename: 039.png

*End frame filename: 091.png

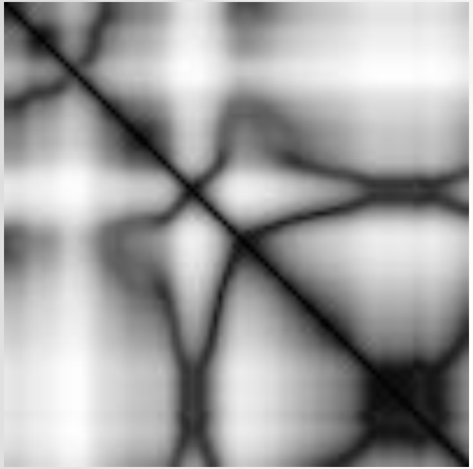
What alpha value produced the smoothest loop? alpha = 0.01

Link to your candle video texture gif:

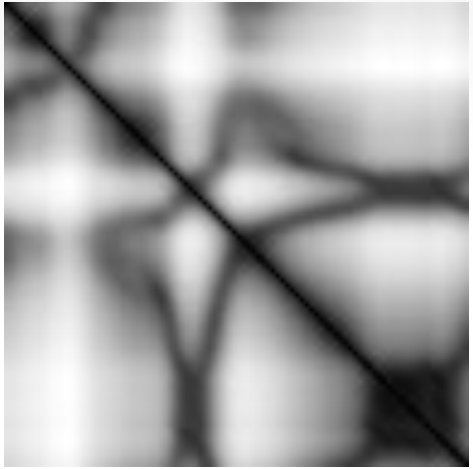
https://drive.google.com/open?id=10dGlyzIX9oKM7nWeANGxOro_i2MwIDV3

* *Frame filename is from original set of frames*

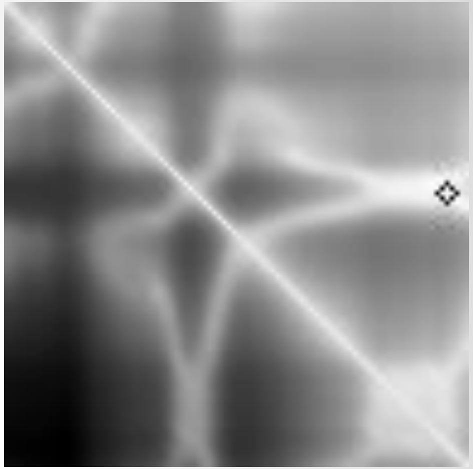
Transition Matrices for Best Sample Conditions



Diff 1: computeSimilarityMetric()



Diff 2: transitionDifference()



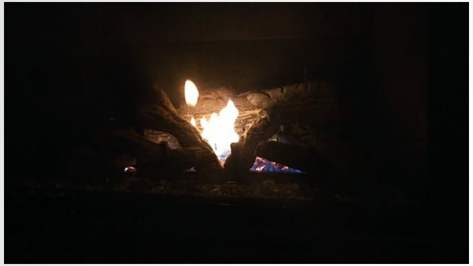
Diff 3: findBiggestLoop()

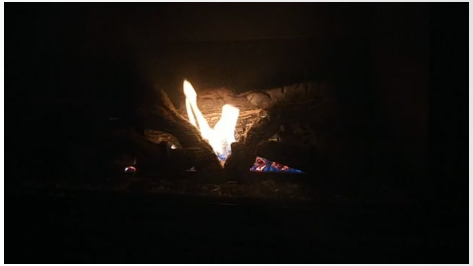
Results from Sample Input

- Clearly explain your procedure to get a good result
  I started with alpha value equal to 1, but this produced not a good transition between first and last frames and diff3 matrix looked like a smooth gradient from bottom left to top right.
  After that I tried decreasing alpha value until program returned just one frame as output and again increasing it until it returned bigger values and pattern on diff3 was similar to one on diff2.
  To get a good result transition point on diff3 should be slightly brighter than diagonal values (alpha is not too small) and should be located in the upper triangle but preferably not in the top-right corner (alpha is not too big).

- What was difficult? Explain.
  The biggest challenge was to find right alpha value to the image. Bigger values lead to bigger loops but not necessarily better results and vice versa smaller values result in better transitions, but loop from one frame is hardly a loop. So understanding diff3 matrix helps a lot, when you can see places of possible transitions, but comparing first and last output frames also helps.

Best Results from Your Own Input





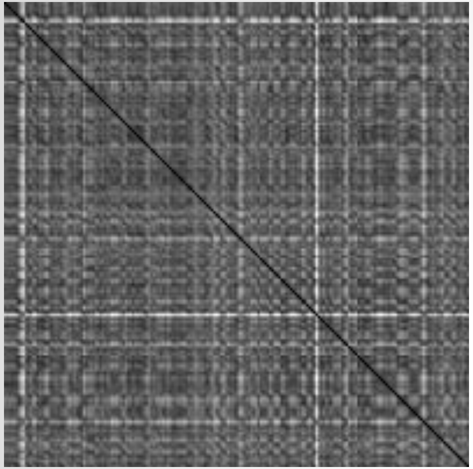
Start frame filename: 019.png
Index frame number: 19

End frame filename: 194.png
Index frame number: 194

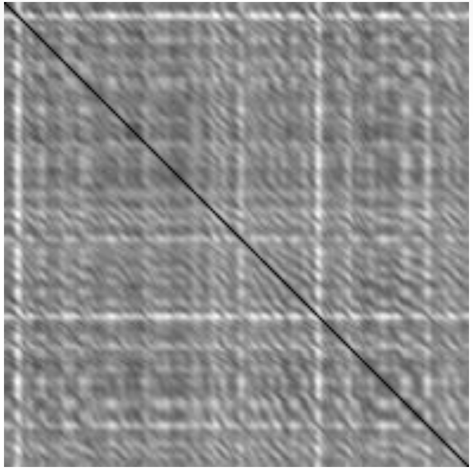
- I took video of the burning fireplace in my house and looped the video using video textures. As a result I got gif of everburning fireplace.

- Alpha value for the best loop - 0.007
- Link to your video texture gif - https://drive.google.com/open?id=1P6jtsNxfNzaKThMo2HeWeNcRoAEEkcdC
- Link to the frames (folder) - https://drive.google.com/drive/folders/1-IJI7-Vgv-IppIMmUkB-q0ld9bgMWjtx?usp=sharing

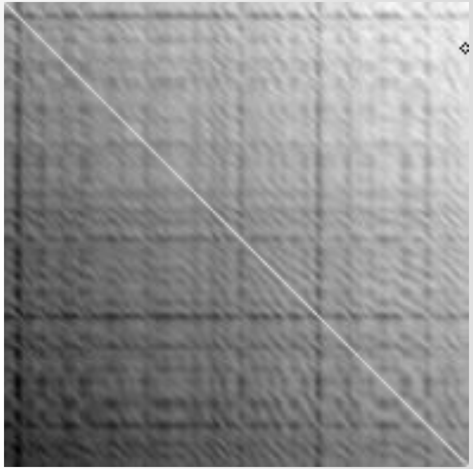
Transition Matrices for Your Own Best Input



Diff 1: computeSimilarityMetric()



Diff 2: transitionDifference()



Diff 3: findBiggestLoop()

Results from Your Own Input

- Did you get a good result on your own video? Explain.
  Yes, I got a satisfactory result. Transition between beginning and the end of the loop is hardly noticeable as first and last frame are actually very close (for fire transition). And loop is quite big.
- How was this video different from working on the sample frames?
  My video has more frames and there are many frames transitions that have similar transition score, so determining best transition spot just based on diff3 matrix was difficult. On the other hand frames were similar enough to not notice transition due to the nature of the fire that changes very fast on its own, so noticing transition on normal speed is very difficult.
- What was difficult? Explain
  Just like in case of sample frames the most difficult part was finding right alpha, especially with more complex/repeating patterns on diff3.
- What would you do differently?
  Next time I would try filming fast changing objects such as fire with higher frame rate to get more smoother transition between source frames, which should give me better idea how good my first and last output frames match each other. For 30fps there are just too many transitions that seems “natural”.

Finding Alpha

- Describe how you determined the best alpha value for each video texture (candle and your own).
  In both cases I started with alpha=1 and decreased it by 10 until I had just one frame as output. This was the indication that my alpha is too low. After that I started to double, triple, etc. that alpha until I had more than one frame. In both cases first returned loops with more than one frame had plenty of frames in them, but if that wouldn’t have been the case I would keep increasing alpha until I had enough frames and hoping that transition would have been still a smooth one.

- How are your results affected as alpha increases? Decreases? (Try changing by orders of magnitude, e.g., x1/10, x10, x100, etc.)
  As alpha increases weight of loop size starts to dominate over transition score and as the result we get bigger loops but bad transition. On the other hand too small alpha means that frames with best transitions will be chosen, but that would probably be just one frame as the transition score is the best when frames are the same. As such alpha should be chosen to strike perfect balance between smooth transition between first and last frames and length of the loop.

- Was the best alpha for your video the same as the one for the sample video? Discuss.
  No, it was smaller since I didn’t have frames as similar as the best pair in sample video. If I would have used more frames (bigger fps) then perhaps alpha would have been bigger. If I had used the same alpha then my loop would have been even bigger, but transition would be worse.

Resources

[1] Video Textures, Scholdl, et al; SIGGRAPH 2000

[2] https://en.wikipedia.org/wiki/Markov_chain

Above & Beyond (input video)

I decided to take my own video to see what challenges might appear from this. Here’s what I learned:

1. Camera stabilization is very important in this case. I took my video on my OnePlus 3T phone and even though video seemed very stable camera movements in fact accumulate very fast with time and as the result transition is clearly visible in the final looped image.
2. Fast changing objects like fire require bigger frame rates to get more smoother transition between even source frames. Resulted gif might still look good if gif also used the same frame rate or higher, but last and first frames match might be much better if I would have had more frames in between source frames.

Above & Beyond (Markov chain generator)

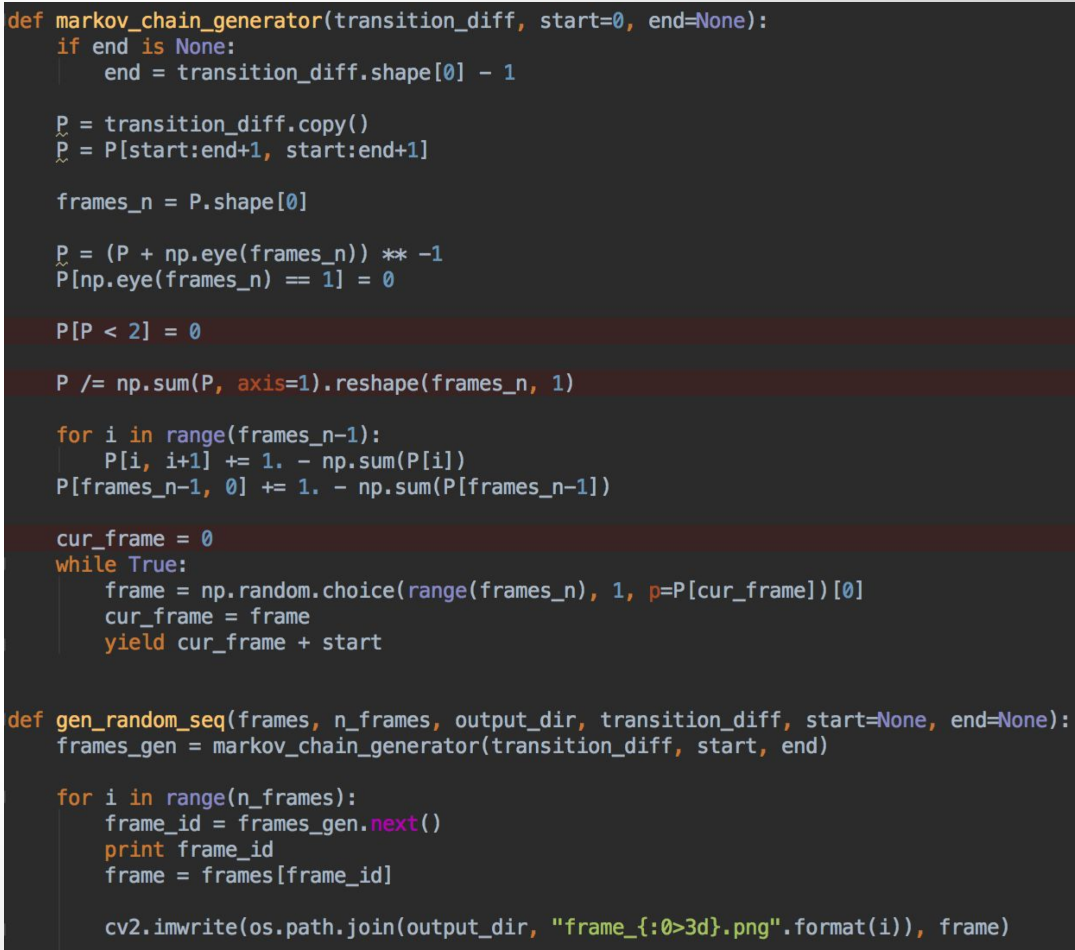
Another thing I wanted to try is to implement Markov chain generator for frames in potentially infinite unlooped video.

The way it works is that I convert transition difference matrix to probability matrix. In this probability matrix each row is the transition probabilities for particular frame. Using random choice with those probabilities as weights Markov generator can randomly create next frame id.

I had to set probabilities that correspond to high transition differences to 0 as well as for all transitions to the same frame, so that video wouldn't get stuck on the same frame and wouldn't jerk too much.

Resulted gif isn't as smooth as using the biggest loop method, but it can generate video without repetitions making it look like a infinite video (with low frame rate).

Result link:
https://drive.google.com/open?id=1TS1-UcNpQYCC8svDFsRae9H8kRp5bXw7



```python
def markov_chain_generator(transition_diff, start=0, end=None):
    if end is None:
        end = transition_diff.shape[0] - 1

    P = transition_diff.copy()
    P = P[start:end+1, start:end+1]

    frames_n = P.shape[0]

    P = (P + np.eye(frames_n)) ** -1
    P[np.eye(frames_n) == 1] = 0

    P[P < 2] = 0

    P /= np.sum(P, axis=1).reshape(frames_n, 1)

    for i in range(frames_n-1):
        P[i, i+1] += 1. - np.sum(P[i])
    P[frames_n-1, 0] += 1. - np.sum(P[frames_n-1])

    cur_frame = 0
    while True:
        frame = np.random.choice(range(frames_n), 1, p=P[cur_frame])[0]
        cur_frame = frame
        yield cur_frame + start


def gen_random_seq(frames, n_frames, output_dir, transition_diff, start=None, end=None):
    frames_gen = markov_chain_generator(transition_diff, start, end)

    for i in range(n_frames):
        frame_id = frames_gen.next()
        print frame_id
        frame = frames[frame_id]

        cv2.imwrite(os.path.join(output_dir, "frame_{:0>3d}.png".format(i)), frame)
```