SIGGRAPH 2015 Course

User-Centric Computational Videography

Christian Richardt  
Intel Visual Computing Institute  
MPI Informatik

James Tompkin  
Harvard University

Jiamin Bai  
Light, Co.

Christian Theobalt  
MPI Informatik

Course website:  
http://gvv.mpi-inf.mpg.de/teaching/uccv_course_2015/
Course syllabus

- Introduction and welcome
  [Richardt, 0:10]

- Background:
  State-of-the-art video tools
  [Richardt, 0:20]

- Timeline editing
  [Bai, 0:20]

- Model-free video editing
  [Theobalt, 0:20]

- Model-based video editing
  [Theobalt, 0:20]

- Break
  [0:15]

- Spatiotemporal video editing and processing
  [Richardt, 0:20]

- Motion editing in videos:
  Cinemagraphs & Cliplets
  [Bai, 0:20]

- Exploring videos
  [Tompkin, 0:20]

- Exploring videos in contexts
  [Tompkin, 0:20]

- Closing and Q & A
  [all, 0:10]
Cinemagraphs are a new type of media that is in-between a photo and a video. It contains motions from a video, but still regions of a photograph. They also loop seamlessly. They are compelling but are tedious to create. An example such as this will take a professional even with sophisticated tools several hours or more to create manually.
The work that cumulates to cinemagraph creation started in the 2000s with Video Textures. Here, they create a seamless looping video by finding good transitions to transition to.

Later, graphcuts was use to sample a short clip to synthesize a seamlessly looping clip by copying appropriate pixels.
One natural extension is to create panoramic video textures from an input video. Here the algorithm uses a graphcut approach to stitch together pixels as it sweeps across the scene to create a seamless loop in both spatial and temporal domains.
If we reconstruct the background explicitly, we can have a variety of motion types. For example, the waterfall and waterwheel is seamlessly looping, while the man walks over the bridge.

The gist is to still create panoramic video textures, but the construction is more sophisticated and allows the initial camera to pan/zoom over the scene arbitrarily, and for multi-layer motions within the scene (pseudo-random background, plus individual elements repeating motions).
The goal is to allow the user to change the viewpoint at will of a pseudo-random motion. This requires a precomputation of per-video similarity matrices, and between-video similarities. We synthesize a new view using real-time optical flow. The effect is subtle, but it works very smoothly. It also allows more creative combinations, e.g., gas hob zoom view while changing gas level.
The first work to introduce cinemagraphs to the community analyses the video to find regions that are suitable for motion looping. The user can then click and select which areas are included in the final result.
Microsoft released a free tool called cliplets that allows users to create moving images. They argue that a moment can be better captured with a series of different motions embedded into the media. Here, notice that the candles are burning. The girl then blows the candles, and the smoke from the candles rises up.

I encourage everyone to download the tool and try it for yourself.
We can further reduce the effort required to create cinemagraphs. Here this project seeks to create automated video loops. However, the user has an option to decide how much motion is included in the final output. If you pay attention to the red bar at the bottom of the video; the length of the red bar indicates the amount of motion included in the final result.
In some cases, de-animating an object allows one to better observe the finer motions within it. In this example, we de-animate the musician of the guitar while retaining the motions of his hands. This allows viewers to better observe the motion of his fingers and potentially makes it easier to figure out the notes he is playing.
Here is how our system works,

For an input video, the user draws green strokes on a regions in a single frame with motion she wishes to remove. These strokes are used to select tracks to guide a spatially varying warp. The fretboard of the guitar no longer moves in the warped video. However, the background now contains unwanted motion. The user then draws compositing strokes to specify regions that should be dynamic in blue, or static in red. Our method then composites the warped video with still frames to generate the final output.
In this example, we de-animate the class, and create a seemingly endless pour of beer.
We can also use the same technique to create portrait cinemagraph. Notice how it loops seamlessly.
In a game of roulette, it can be difficult to understand and predict the path of the steel ball as there is a counter spinning inner wheel. Pause. Our method can deanimate the inner wheel, allowing the user to better see the simple path the ball takes with respect to it.

Pause

We draw de-animate strokes on the inner wheel and dynamic strokes on the inner wheel and ball. Static strokes are drawn on the background and edge of the wheel.

This results takes approximately 10 mins to compute
We can also use this technique to edit videos. Here we are looking at a deanimated representation. Notice how the glass is stationary. When we make an appearance edit to the glass, it appears natural and plausible. We can then propagate this edit by unwarping the video. This allows us to make appearance edits on one frame and automatically propagate them in the video.
We often have still portraits for our online webpages. Wouldn't dynamic portraits better convey our personalities?
The previous approach allows high quality portrait cinemagraphs to be created. However, it is slow and requires user input to generate results..
However, we can leverage on computer vision techniques such as face detection, facial feature detection to automatically stabilise the subject and composite motions into the final results. This automatic approach produces high quality results with up to 60x speed up compared to the previous approach.
Here is a video captured with a mobile device. The large sways in the motion of the camera results in the video that is shaky and undesirable.
We can stabilize the video using state of the art techniques such as Bundled Paths. The stabilized video is much better; however, there are still sways in the build façade and the subject. This is because the algorithm does not have any semantic knowledge about the scene during the stabilization process.
We can use user input to improve the result by describing how the output should look like. Here is the user-assisted video stabilization result. Notice that the façade are mostly upright now.
Here is how our system works. We start with our implementation of bundle paths. One reason why the stabilized video is not satisfactory is because tracks used to guide the warp do not belong to the background. In our work, we cluster tracks based on their trajectories and allow users to modify track selection by toggling the track clusters using the mouse. Here, brightly colored tracks are selected and grey tracks are pruned. Notice how the user removes tracks on the dynamic objects in the video.

Here we show the input video stabilized with only background tracks. Notice while the output is improved over the initial result, the background and the walls still move and sway. This is because the algorithm does not have any semantic information about the video.

To remedy this, we allow the user to specify how regions in the output video should look. The user draws quadrilaterals and deforms the region by dragging the corners of the quad. The region is remapped in real-time to provide user feedback. We then solve for a stabilized output where the user provided region constraints are met.

This is our final result. Notice the background and walls do not sway as much.
Here is the comparison of the results. Notice that the facades are now mostly vertical.
Here is another result. Notice that the rotation at the end of the video is now corrected.
Conclusion

- Motion editing is hard
- User-intent coupled with specially designed algorithms can create high quality content.