Animating Pictures with Eulerian Motion Fields

Paper Review – Noah Hendrickson, Eric Trempe, Louis Wang

Summary

Simulating realistic motion is notoriously difficult due to the variety of different scenarios and environmental factors in play. Aleksander Holynski, Brian Curless, Steven Seitz, and Richard Szeliski propose a 'fully automatic method for converting a still image into a realistic animated looping video' in the paper we reviewed, titled "Animating Pictures with Eulerian Motion Fields." General motion is complex, especially inferring from a still image, so the scope of the motion generated by the model is limited to fluid motion. Previous work in this field often relies on video data as an input to generate the output video loop, which makes the solutions less widely applicable. Others use methods which produce suboptimal results such as backward flow or recurrence resulting in dull or distorted results. The model described in this paper shows to be a suitable improvement to previous models in the space by using forward motion prediction on still images.

The model looks to produce a looping video consisting of *t* frames from a still image using a dual forward-backward motion prediction model. It is split into two parts: a motion estimation component using an image-to-image translation network to generate the eulerian flow field (**M**) which is used to generate displacement fields for every frame ($\mathbf{F}_{\mathbf{o}\rightarrow\mathbf{t}}$, $\mathbf{F}_{\mathbf{o}\rightarrow\mathbf{t}\text{-}\mathbf{N}}$), and an animation component utilizing an encode-decode schema where the input frame ($\mathbf{I}_\mathbf{t}$) is encoded into deep features ($\mathbf{D}_\mathbf{t}$), which are warped using symmetric splatting ($\mathbf{D}_{\mathbf{t+1}}$) to decode the next frame ($\mathbf{I}_{\mathbf{t+1}}$*)*. Each pixel in the input image corresponds to an (x,y) coordinate in the image space and is annotated with the (x,y) velocity vector at that location.

The image-to-image translation network is trained on color-motion pairs and only needs to generate the flow field once in order to be used to find all displacement fields. Eulerian integration is applied recursively such that the displacement of every pixel can be found for every frame. The animation model is a bit more complex and uses a slightly modified version of another encoder-decoder model used for generating different viewpoints of an image but with the view changing method replaced with a novel splatting technique. With normal splatting, warping deep features using displacement maps results in a tendency for pixels in the image to cluster towards the bottom of their motion. For example, water pixels will cluster towards the bottom of a waterfall leaving the top area unknown, which forces interpolation to be done in the unknown area and resulting in an uncanny, stretched texture. To combat this, the paper utilizes the idea that the process to find the forward motion will be exactly the same as the backward motion, just in reverse. The negative of M is found and used to compute $F_{0\rightarrow\text{-}t}$ to produce $D_{\text{-}N}\ldots D_0$ where the last of the warped features is the same as the initial features input. Another problem arises when multiple pixels are mapped to the same location. To disentangle the effects of multiple colliding pixels, softmax splatting with a learned metric Z is used to estimate each individual contribution.

The animation model also handles the looping of the video by making use of the assumption that the end of the $F_{0\rightarrow\text{-}t}$ mappings is identical to the beginning of the $F_{0\rightarrow t}$ mappings. A modification to the softmax is made such that the former contributes nothing at the start but fully at the end, with intermediate contributions controlled by a scaling coefficient. For training, the paper presents a method that circumvents the non-looping characteristic of the training videos. Two frames are taken, one at the beginning and one at the end, and the model effectively learns to interpolate between the two. During testing, the model would effectively interpolate between the input frame and itself, resulting in a looping video. After training, the model consistently provided better results to other comparable models. Detailed statistics can be found in section 7 of the paper.

Critique

"Animating Pictures with Eulerian Motion Fields" provides a good overview and description of the objective of the model, the model itself and the new ideas implemented within, and the improvements over previously designed models for similar tasks. The processes are relatively easy to follow, even if you don't have a lot of experience with the underlying architecture. The point of the paper is presented in a logical order that helps bolster understanding, as the authors guide the readers through the model in the order that the data flows. It also does a great job showing how this particular project has advanced the field and how it improves upon other prior studies, providing sufficient justification for novelty. The set-up and goals are clearly expressed, and it is easy to understand the applications of this paper.

However, there were a few points that we think could have improved the paper and led to a better understanding of the model and its applications as a whole. Firstly, an observation that we made was that the learning rate for the discriminator was always higher than that of the generator, in a sense creating an environment where the discriminator was 'leading' the generator. Though this may be more of an issue with generative adversarial models in general, we think it would have been interesting as a learning exercise to explore the results if the roles are flipped, or if multiple discriminators were used at the same learning rate, notwithstanding the added training cost in such a scenario.

Additionally, we would have liked to see a bit more exploration into extending the solution into three dimensions. A cursory glance of the proposed solution seems like such an extension would not be too difficult, and might even make some of the motion seem a bit more natural if the depth component was included. Another issue was that the range in velocities portrayed seems narrow. Many of the images appeared to move at the same rate, even if depicted objects or textures would appear more realistic with a wider velocity range. Adding an acceleration component to the model might improve upon this to make, for instance, smoke and water move differently than one another, as the motion of smoke in the atmosphere behaves differently than water in freefall. Adding acceleration may require more advanced integration methods, such as Heun's method, and may make some of the simplicities of this model, such as reusing M and symmetric splatting, harder to implement. In the end it may just be a tradeoff between realism and ease.

In the evaluation portion of the paper, there was quantitative analysis of the model's performance for most elements, comparing pixel positions for 60 frames with the frames of the ground-truth video. But even if every still image selected closely matched with the true image there can still be errors with the motion connecting the two images. For instance, since the original video does not loop, forming a convincing loop may involve sacrificing pixel-wise similarity to the ground-truth video near the looping frames. To supplement quantitative similarity measures, the authors of the paper conduct a user study to evaluate the overall performance. However, only a realism ranking of various competing methods was given from the study, and a score range of 0 to 3 may not be nearly enough to judge quality with sufficient granularity. Qualitative analysis focused solely on the positives rather than negatives. We felt that some of the images looked somewhat unnatural, and would have appreciated further reflections on qualitative faults, causes of these faults, and further work that can be done to mitigate these issues. For example, the motion of one of the waterfalls never reached the water below. Was this a result of limits on loop duration, or limits of predictive capabilities? Finally, while it is clearly stated that this model is meant only for fluid motion like waterfalls, insights into how the model could be adjusted to work for non-fluid motion would have been appreciated. In short, the main deficit was the lack of discussion regarding its faults, leaving no discussion for future work.