Per-Object Motion Blur for Live Action

Henry Huang, Ivan Guo, Nancy Makar, Ivan Guo

Table of Contents:

Project Description:

As videos are just rapid streams of pictures shown one after the other, “smoothness” has long been desirable. Traditional methods include recording at higher frame rates and motion blur. Higher frame rates require foresight, and cannot be appended. Thus, if someone wants a smoother video, they have to rerecord it, which is not always feasible.

Regular motion blur is cheaper but affects entire frames in a video. In ideal conditions, like with a stable camera, this achieves the desired effect of smoothness. However, if the camera moves, then everything in the frame is moving relative to the camera. This causes background objects to be blurred, which can harm visual clarity.

This project seeks to implement an improved version of motion blur, known as per-object motion blur. This method only blurs objects that are moving irrespective of the camera’s movement. For example, if a camera turns while pointing at a person waving, regular motion blur would blur the entirety of the frames for when the camera is turning. Per-object motion blur would instead only blur the person’s arm waving. Consequently, none of the details of the unmoving person or background would be lost, resulting in more smooth motions while preserving visual clarity.

Per-object motion blur is usually applied to animation, where it is easy to identify beforehand which objects are moving. This project seeks to implement it in the context of live action recordings, where moving objects cannot be determined beforehand using an object tracking machine learning model.

Walkthrough Example:

Component Description:

The first component of this application is the webpage. On it, a user can upload any video file to have the per-object motion blur effect applied. Section 1 of the diagram.

After receiving a video from the webpage’s frontend, the next component detects and returns relevant objects in motion. The first step that this component does is run a Pytorch, Faster RCNN model on each frame of the video, where each frame is obtained using OpenCV. The Faster RCNN model performs object detection for 16 classes that are expected to be in motion. For example, forearms and cars. The bounding box results from the model are then passed to the SORT algorithm. The SORT algorithm turns the object detection results into motion/object tracking results by assigning a unique ID to each tracked object throughout the video, while whatever SORT detects as the same object from a previous frame keeps its old ID. At the end of this component, there is a list of tracked objects for each frame in the video that gets passed to the next component. Section 2 of the diagram.

After receiving a list of detections with their associated tracking IDs, this component seeks to interpret and process the data to produce blurred images. For every frame, this component goes through all the detected objects in the frame. For each detected object, it checks the next

frame for if there is a detected object with a matching ID. If not then this component will do nothing. Otherwise, it takes the two bounding boxes of the matching objects and attempts to create a new frame in between the two that smooths the motion. It does this by translating the detected object in the first frame to a point in between where it is in the first and second frame, and then blurring it. The intensity of the blur is determined by the velocity of the object that is tracked. The velocity is calculated by taking the displacement between the two objects in pixels, adjusted for image size, and dividing it by the frame time, which is the time between frames. Section 3 of the diagram.

With the new inserted frame from the previous component, the new frame is then inserted between the two original frames. Then, with all the original and new frames, they are combined with OpenCV to create a new, continuous video. Section 4 of the diagram.

The final component returns to the web page, where it plays a preview of the now processed video. Additionally, it gives the user the option to download it. Section 5 of the diagram.

Diagram: