Novel Methods for Motion Vector Estimation

Proposal

In typical video content, frames typically are highly temporally correlated, allowing for reuse of information from previous and future frames to construct the current frame. This techinique is used in several mainstream video compression algorithm specifications such as h.264, h.265, and VP9.[1][3][2] In these algorithms, motion estimation is used to create transfer functions from temporally nearby frames for mapping to the current frame. However, this mapping is often insufficient to fully reconstruct the frame, resulting in the need to pack additional information for each frame. The efficiency of this kind of video compression algorithms can be improved using more advanced temporal interpolation algorithms that provide better estimations of the current frame. The goal of this project is thusly to implement an algorithm utilizing machine learning techniques for temporal interpolation.

An interesting application of temporal motion estimation is that it can also be used to extrapolate additional information from existing video, also known as temporal upscaling. Existing packages such as MVTools and SVP utilize the motion vector estimation algorithms of h.264 to generate new frames, however, as these algorithms do not target a complete reconstruction of the target frame, a significant amount of masking needs to be done to remove artifacts.[5] A machine learning assisted algorithm may be able to improve upon existing implementations to allow for temporal upscaling. This also provides a tangible reference to judge the performance of the developed algorithm in this project.

The difficulties associated with such an algorithm arises from the computational limitations of current machines. A fairly common 3840x2160-30p 3-channel video will require processing 746 million entries per second. A full exhaustive motion search requiring approximately $10^{10^{9.8}}$ calculations is entirely infeasible. However, convolutional neural networks can efficiently reduce and extract information from image data, and may be applied to motion vector estimation, as well as directly constructing the target image.

Datasets for this project is easily generated from existing video, by simply copying frames at half the temporal resolution (skipping even numbered frames). The odd numbered frames will be used to generate the input features, and the outputs will be either motion vectors to generate the even numbered frames or a direct reconstruction of the even frames. An enourmous amount of data is available from online media, and can be generated using mobile devices' cameras.

As manual feature generation for image data is exceedingly difficult, the machine learning models considered will require some feature generation capabilities. Initial development will involve modified variants of the resnet architecture, although this may change throughout the project depending on early results.[4] The generated features will be linearly mapped to a grid of motion vectors.

Due to the need of highly flexible network architectures and likely a significant amount of image preprocessing, PyTorch was selected for its ability to propagate gradients through arbitrary linear operations outside of a network class and its ability to perform operations efficiency on a GPU using CUDA.[6] Yuke Wang also has existing frameworks written for machine learning using PyTorch.

References

[1] Advanced video coding for generic audiovisual services. International Telecommunication Union, 2021.

[2] Grange, Adrian, et, al. VP9 Bitstream & Decoding Process Specification. Google, 2017.

[3] High efficiency video coding. International Telecommunication Union, 2021.

[4] He, Kaiming, et, al. Deep Residual Learning for Image Recognition. IEEE, 2015.

[5] Manao, et, al. MVTools. Doom9, 2018.

[6] PyTorch Documentation. Torch Contributors, 2019.

Team

Team Name: Vestaia

Members

Yuke Wang - 12280359