Real Time Video Compression with 3D CNNs and Autoencoders [Milestone]

Brennan Shacklett
Stanford University
bps@cs.stanford.edu

1. Introduction

Video compression is an important area of research due to the ever increasing resolution demands of video, as well as the rise of relatively new media such as live streaming. Unfortunately, video codecs and their corresponding compression algorithms typically make use of a large set of finely tuned heuristics, which can make developing codecs and testing new features quite difficult. Given the success of neural networks in discovering approximations and heuristics in other fields, such as work on image super-resolution [1], it seems natural to apply these techniques to video compression.

To somewhat refine the problem, this work is specifically interested in exploring improving real time video compression: compression without knowledge of future frames and able to operate with only $\frac{1}{60}$ second alloted to each frame. This precludes use of some techniques in traditional video encoding such as 2 pass video encoding, which significantly complicates the compression pipeline [4].

Since video compression is a complex area of active research, this project aims to find neural network based strategies for arguably the 2 most important parts of current video codecs: intra-prediction and inter-prediction, which correspond to reconstructing a video frame without relying on previous frames, and constructing a video frame from previous reference frames respectively [2]. For the problem of intra-prediction, this project will attempt to use autoencoders, possibly combined with super-resolution techniques, and inter-prediction will be achieved with a combination of LSTMs (such as used in PredNet [3]) and 3D convolutions.

2. Problem Statement

The problem is how to reduce the bitrate required for a video sequence, while minimizing distortion from the source material as much as possible. This tradeoff is known as rate-distortion optimization, and will be the metric used for evaluation. Specifically, the rate-distortion curve of this project's video compression scheme will be compared to the rate-distortion curve of MPEG-1. While MPEG-1 is a very old codec, it contains most of the key features used by modern video codecs, so if a neural network approach can achieve better rate-distortion properties, it will suggest that with further work similar approaches may be able to beat state of the art codecs. As a point of reference the rate-distortion curve of AV1, a state of the video codec, will also be included. The expected result is that the neural network approach will be able to perform better than MPEG-1, but likely worse than AV1.

The network will be trained on the objective-2 dataset from the Xiph foundation, and video codecs will be compared on objective-1, as well as some assorted longer sequences from the Blender project. If a large amount more data is required for training, datasets such as Moments will be investigated; however, using videos other than raw source material for video codec evaluation is generally not recommended within the community (and the Moments dataset is compressed).

There are several different metrics available for measuring the distortion of a video codec, some of which use neural networks to achieve better perceptual understanding of visual quality. Unfortunately, most comparisons in the video codec community are restricted to PSNR (which corresponds to the L2 norm), and SSIM – an improvement over PSNR that also accounts for variance. For the purposes of this project SSIM will be used as the measurement of distortion as well as the loss measurement for training parts of the network, unless performance becomes an issue, in which case the simpler PSNR will be considered.

3. Technical Approach

For both intra-prediction and inter-prediction, the current frame is divided up into 64x64 blocks, with padding in the bottom and right blocks if necessary. This allows supporting arbitrary video sizes and significantly reduces the input size to the neural network. Larger blocks would likely be more desirable for higher resolution videos, and in fact modern codecs support variable block sizes, but this seems difficult given a neural network architecture where the first convolutional layer expects a certain number of arguments.

Intra-prediction will occur with an autoencoder designed to substantially reduce the dimensions of the block. The decoder portion of the autoencoder will pass into a network modeled after the super resolution network from [1]. Since the cited super-resolution network requires the low resolution version to first be upsampled, the idea is the autoencoder will provide the equivalent of the upsampled low resolution version of the image and the super resolution network will attempt to correct for some of the visual information lost in the autoencoder.

Inter-prediction is the more interesting part of the project, with more variables in the technical approach. The first approach will be a simple 3D convolution for each block which uses the previous 10 blocks to predict the next block. A small extension to this will be for the 3D convolution to accept a larger input block than output block: for example a 192x192 input to give the network a view of the surrounding blocks for reconstruction.

The second approach if the 3D convolution turns out to be too simplistic will be to use an LSTM, perhaps with a similar structure to [3]. This should allow for more intelligent selection of reference frames, as well as for information from previously reconstructed blocks in the current frame to be used. Using a 3D convolution would be preferable however, since it would allow for more parallelized reconstruction, and may be simpler for training.

After Intra-prediction and Inter-prediction are both completed for a given block in the video, whichever reconstruction method has a higher SSIM will be selected. A bit encoding which mode was selected is then written out to disk, along with the reduced dimensions from the autoencoder if intra-prediction was selected. If SSIM is below a certain threshold a truncated version of the residual with the source block will also be stored. These bits will then be passed into an ANS encoder to gain some compression savings and attempt to be competitive with modern video codecs.

4. Preliminary Results

Thus far work has focused on the next frame prediction part of compression. This step is equivalent to inter-frame prediction in traditional codecs, and is typically the largest source of compression savings for video compression, so it is arguably the most important part of the project. One important point is that although it may be tempting to train a network to reconstruct the Nth frame using the previous M source frames, in lossy compression the M previous source frames are of course not available during decoding. Instead, only the lossy reconstructions of the M previous frames are available, which means the reconstruction process must be careful to avoid errors compounding. Unfortunately, accurately handling this complication requires that the entire codec and reconstruction process be fleshed out; therefore the preliminary experiments were conducted simply using the source frames in the hope that the techniques will ultimately transfer over.

Initially all training was conducted on 1920x1080 videos, where the network attempted to predict the full next frame. While this strategy may be promising with more data, with the relatively small amount of source material available the network simply was unable to predict frames in any meaningful way. One solution to this may be preprocessing the source material with reflections, color shifts etc, but this hasn't been explored yet. Instead, as described in the previous section a block based system was used, where each block in the video is treated totally independently. Unsurprisingly, while this achieved much better results, since the network for a block is significantly smaller and there are significantly more effective training examples, there are significant block artifacts (discontinuities at block edges). This is also an issue with traditional video codecs, and is typically fixed with a smoothing algorithm often referred to as a loop filter. Undoubtedly some neural network based denoising filter could be applied with similar success, but a strategy that somehow factored discontinuities into the loss function to avoid them entirely would be preferable. This may be possible with an LSTM where previously constructed blocks in a given frame (perhaps the upper and left blocks) are part of the internal saved state, but this may come at too much of a cost to parallelism and trainability.

The final technique that has been experimented with up to the milestone is how to select reference frames. While simply using the previous M frames is tempting from a simplicity standpoint, current video codecs typically have multiple reference frames that are selected somewhat more intelligently: perhaps preserving a background shot from many frames prior where action will take place somewhat later in the video. Unfortunately all techniques experimented with produced worse results than simply selecting the previous M frames: preserving every 100th frame etc. Ideally this would be optimized for by an LSTM that was able to select its own reference frames; however this is future work.

References

[1] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *CoRR*, abs/1501.00092, 2015.

[2] D. Le Gall. Mpeg: A video compression standard for multimedia applications. *Commun. ACM*, 34(4):46–58, Apr. 1991.

[3] W. Lotter, G. Kreiman, and D. D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. *CoRR*, abs/1605.08104, 2016.

[4] P. H. Westerink, R. Rajagopalan, and C. A. Gonzales. Two-pass mpeg-2 variable-bit-rate encoding. *IBM Journal of Research and Development*, 43(4):471–488, July 1999.