

# Related Work

Austin Dworaczyk Wiltshire

In terms of published academic work, there are not a lot of currently available papers that detail a distributed encoding system. A simple search for “distributed video encoding” often brings up an alternative query “distributed video coding” which is another form of video compression that takes the brunt of the work away from the encoder and places it on the decoder. Again, it has little to do with the desired subject of distributed video encoding, so discovering published works in this area has proved almost fruitless.

Nonetheless, there are a few pieces of related work, one of which is written by Rafael Silva Pereira and Karin K. Breitman, titled *Video Processing in the Cloud*. The algorithm they developed is a modification of the MapReduce technique, and is quite a reasonable design. It is based on the Split and Merge principal, with a video being divided into chunks that match up with the GOP (Group Of Pictures) size of the input video. A video can only be split on it’s I-frames, or key-frames, as they are the main references for the rest of a video. It is because of this inter-frame dependence that the standard MapReduce distributed problem model won’t work, as MapReduce does not know how to discriminate when splitting a source material, in addition to maintaining synchronicity between the video portion of an input clip and its associated audio portion.

So, by splitting along the I-frames, encoding chunks can be made. The problem stated by Pereira and Breitman, is determining optimal chunk size. Several factors need to be taken into account when splitting a video. First, the number of available computing resources is a big factor. Second, the overall length of the video is very important. Third, the desired encoding time is a key factor, along with the availability of a buffer or queue of un-encoded and encoded segments. This is one area of research that my thesis will focus on.

Other than that, there is little work that has been done in this field, at least in terms of published work. There are several companies that likely implement a system like this, such as YouTube and Zencoder, but their systems are proprietary and closed to the public. In addition, no specifications or system design have been published through their work. There is an obvious reason for this, as they would be releasing the standards directly to their competitors. So, currently there are not a wide variety of publications or implementations of this system, which makes further research a bit difficult.