P-frame Coding Proposal by NCTU: Parametric Video Prediction through Backprop-based Motion Estimation

Yung-Han Ho[1] Chih-Chun Chan[1] David Alexandre[2] Wen-Hsiao Peng[1,3] Chih-Peng Chang[1]

{hectorho0409.cs04g@, dororojames.cs07g, wpeng@cs., cpchang.cs08g@}nctu.edu.tw

[1]Computer Science Dept., [2]Electronics Engineering Dept.,
[3]Pervasive AI Research (PAIR) Labs, National Chiao Tung University, Taiwan

Abstract

This paper presents a parametric video prediction scheme with backprop-based motion estimation, in response to the CLIC challenge on P-frame compression. Recognizing that most learning-based video codecs rely on optical flow-based temporal prediction and suffer from having to signal a large amount of motion information, we propose to perform parametric overlapped block motion compensation on a sparse motion field. In forming this sparse motion field, we conduct the steepest descent algorithm on a loss function for identifying critical pixels, of which the motion vectors are communicated to the decoder. Moreover, we introduce a critical pixel dropout mechanism to strike a good balance between motion overhead and prediction quality. Compression results with HEVC-based residual coding on CLIC validation sequences show that our parametric video prediction achieves higher PSNR and MS-SSIM than optical flow-based warping. Moreover, our critical pixel dropout mechanism is found beneficial in terms of rate-distortion performance. Our scheme offers the potential for working with learned residual coding.

1. Introduction

The past few years see some success in learning-based image compression. It can now perform comparably to modern image codecs, such as BPG, although its complexity remains an open issue. Recently, there emerge few early attempts at learning video compression [2, 3] end-to-end, to address the even more challenging problem of ever increasing video bandwidth.

Like conventional approaches, most learning-based video codecs perform motion-compensated temporal prediction, followed by residual coding. At the encoder side, they estimate optical flow between the reference and the target frames, with the quantized latent representation of the flow map sent to the decoder as additional side information. Due to the lossy representation, the optical flow can only be recovered approximately to warp backward the decoded reference frame in forming a prediction of the target frame. The residual between the motion-compensated frame and the target frame is then separately compressed using learned residual coding.

Although showing interesting performance as compared to conventional codecs, like AVC/H.264 and HEVC/H.265, learning-based video codecs often suffer from having to signal a large amount of motion information, especially when it comes to low bit-rate coding. Moreover, the single-hypothesis prediction nature (i.e. each target pixel is predicted from a single pixel in the reference frame) of flow-based motion compensation is susceptible to compression quality of optical flow. To improve temporal prediction, Ren *et al*. [3] introduce hierarchical bi-prediction with quality layers. To reduce motion overhead arising from bi-prediction, they further derive motion information for bi-prediction from that of uni-prediction, a technique often used for optical flow-based frame interpolation. The use of bi-prediction however incurs additional frame buffering and processing delay.

Recognizing that a compromise between motion overhead and prediction quality must be made, we propose a backprop-based motion estimation scheme. We identify and transmit motion vectors for only few critical pixels in the target frame. This is followed by frame warping using parametric overlapped block motion compensation (POBMC), a classic, multi-hypothesis prediction scheme. In particular, a dropout probability is learned for each critical pixel to strike a better balance between motion overhead and residual energy. One striking feature of our approach is that we view the determination of critical pixels and their motion vectors for each video frame as an optimization problem rather than a learning problem. Their values are optimized explicitly based on minimizing a loss function through the steepest descent and backprop.

We demonstrate compression results with HEVC-based residual coding. As compared to flow-based frame warp-

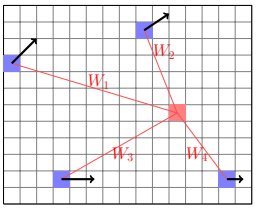

Figure 1. POBMC on a sparse motion field composed of 4 critical pixels (blue boxes) and their motion vectors (black arrows).

ing, our parametric video prediction with sparse motion achieves higher PSNR and MS-SSIM on CLIC validation sequences under the P-frame prediction structure. Moreover, our critical pixel dropout mechanism is found beneficial together with residual coding in terms of rate-distortion performance.

The remainder of this paper is organized as follows: Section 2 overviews our parametric video prediction. Section 3 details our motion estimation scheme, with section 4 elaborating on the HEVC-based residual coding. Section 5 presents experimental results. Section 6 concludes this work.

2. Parametric Video Prediction

At the heart of this proposal for P-frame coding is the parametric overlapped block motion compensation (POBMC) [1], a classic video prediction technique that forms a multi-hypothesis prediction of every pixel s in the target frame I_t by using a handful of sparse motion vectors. Consider the example in Fig. 1, where we have a sparse motion field composed of 4 critical pixels $\{s_i\}_{i=1}^4$ in the target frame I_t along with their motion vectors $\{v(s_i)\}_{i=1}^4$ (see the blue dots and black arrows). In predicting the value $I_t(s)$ of a pixel s in the video frame I_t, POBMC computes a weighted sum of four hypotheses $\sum_{i=1}^4 w_i I_r(s + v(s_i))$, each being a motion compensated signal $I_r(s + v(s_i))$ derived from the reference frame I_r using the motion vector $v(s_i)$ associated with one of the four critical pixels $\{s_i\}_{i=1}^4$. The optimal weights $\{w_i\}_{i=1}^4$ are computed so as to minimize the prediction residual at s in the mean-squared error sense:

$$w_i = \arg\min_{w_i} E\left[\left(I_t(s) - \sum_{i=1}^4 w_i I_r(s + v(s_i))\right)^2\right], \quad (1)$$

subject to $\sum_{i=1}^4 w_i = 1$. Under some signal assumptions, the optimal weight w_i are computed in closed-form to be inversely proportional to the Euclidean distance $r(s; s_i)$ between the predicted pixel s and its surrounding critical pixels s_i [1]:

$$w_i \propto \frac{1}{r(s; s_i)}, \quad (2)$$

where α is a signal dependent hyper-parameter.

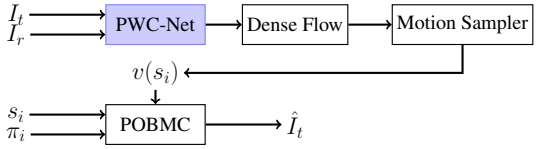

Figure 2. Optimization of the critical pixels $\{s_i\}$ and their dropout probabilities $\{\pi_i\}$.

3. Backprop-based Motion Estimation

In the previous section, we make a strong assumption that we know in advance the locations $\{s_i\}$ of the critical pixels and their motion vectors $\{v(s_i)\}$. We now describe how they are obtained through back-propagation and optical flow estimation. Our task is to determine K pairs of critical pixels and their motion vectors $\{(s_i; v(s_i))\}_{i=1}^K$, in an attempt to minimize the prediction residual between I_r and I_t. In particular, for each of these K critical pixels, a probability value $\pi_i \in (0; 1)$ is used to decide further which critical pixels and their motion vectors (among those K initial candidates) should be compressed and involved in the process of POBMC. In our current implementation, both s_i and π_i are parameters to be optimized via the back-propagation of a prediction loss while $v(s_i)$ are estimated by PWC-net [4], a pre-trained optical flow estimation network. The retained $\{s_i; v(s_i); \pi_i\}$ are compressed losslessly and sent to the decoder.

3.1. Method Overview

Fig. 2 presents an overview of our scheme for determining $\{s_i; v(s_i); \pi_i\}_{i=1}^K$. The process begins with the estimation of an optical flow map F describing the motion for warping backward from I_t to I_r–i.e. $I_t(s) \approx I_r(s + F(s))$. Here we use the pre-trained PWC-net for flow estimation. Because PWC-net yields a flow map that is one-sixteenth the size of I_t and I_r, it is interpolated bi-linearly to full-size. With this dense, full-size motion field F, the motion sampler takes its samples at critical pixels $\{s_i\}_{i=1}^K$, giving rise to $\{v(s_i)\}_{i=1}^K$ (Section 3.2). The predictor $\hat{I}_t(s)$ for a pixel s in the target frame I_t is then evaluated as the weighted sum $\hat{I}_t(s) = \sum_{i \in \mathcal{N}(s)} w_i I_r(s + v(s_i))$, where $\mathcal{N}(s)$ refers to the four critical pixels s_i nearest to s. We then formulate a loss function $L(\cdot)$ (Section 3.3) taking into account the difference between the target frame I_t and its prediction $\hat{I}_t$, together with π_i, the contribution of each critical pixel s_i to the prediction of I_t via POBMC. In turn, the loss function is minimized via the *steepest descent* on π_i and s_i. The process of determining $\{s_i; v(s_i); \pi_i\}$ can collectively be thought of as a form of sparse motion estimation.

3.2. Motion Sampler

In our scheme, the coordinates $(s_i^{(x)}; s_i^{(y)})$ of a critical pixel s_i are continuous variables, with their values bounded from the above by the width W and height H of

the input video, respectively. That is, $s_i^{(x)} \in [0; W]$ and $s_i^{(y)} \in [0; H]$. It is however noted that the dense flow map F is defined only on integer-pixel positions $s = (s^{(x)}; s^{(y)})$ where $s^{(x)}; s^{(y)} \in Z$. Therefore, for a critical pixel at a sub-pixel position s_i, its motion vector $v(s_i)$ is interpolated bi-linearly between those $F(s)$ at integer-pixel positions s:

$$v(s_i) = \sum_s F(s) K(s_i; s); \tag{3}$$

where the bi-linear interpolation kernel $K(s_i; s)$ is defined as $K(s_i; s) = max(0; 1 - |s_i^{(x)} - s^{(x)}|) max(0; 1 - |s_i^{(y)} - s^{(y)}|)$.

3.3. Critical Pixel Dropout

As we indicate previously, a probability value $\alpha_i \in (0; 1)$ is attached to each selected critical pixel to identify which of them need to be communicated to the decoder. This is implemented as an automated mechanism to strike a balance between the overhead for signaling motion information and the reduced residual energy. At test time, only the critical pixels with their α_i exceeding a pre-defined threshold are signaled.

To determine $\{\alpha_i\}_{i=1}^K$ via back-propagation, we re-parameterize it as $\alpha_i = \sigma(\beta_i)$, where $\sigma(\cdot)$ is the sigmoid function and takes as input the parameter β_i to be optimized. Because now the critical pixel s_i and its motion vector $v(s_i)$ has a $\sigma(\beta_i)$ probability of being present for POBMC, the expected value of the predictor $\hat{I}_t(s)$ is evaluated as

$$\hat{I}_t(s) = \sum_{i \in N(s)} \sigma(\beta_i) w_i I_r(s + v(s_i)); \tag{4}$$

for which we further impose the unit gain constraint $\sum_{i \in N(s)} \sigma(\beta_i) w_i = 1$ to ensure that the value of $\hat{I}_t(s)$ will not be blown out. Using Eq. (4), we minimize the mean of the squared prediction error between $I_t(s)$ and $\hat{I}_t(s)$ over all the pixels $s \in I_t$ in the target frame I_t subject to the unit gain requirement by minimizing

$$L_{pred}(\{\beta_i\}; \{s_i\}_{i=1}^K) = \frac{1}{N} \sum_{s \in I_t} \frac{(I_t(s) - \hat{I}_t(s))^2}{\sum_{i \in N(s)} \sigma(\beta_i) w_i}; \tag{5}$$

with respect to $\{\beta_i\}; \{s_i\}_{i=1}^K$. In particular, to reduce the number of motion vectors to be sent to the decoder, we additionally require that only few α_i should be non-zero. This is achieved by regularizing the determination of α_i with the sparsity constraint $\sum_{i=1}^K |\alpha_i| = M$. As a result, our loss function for motion estimation can be expressed as

$$L(\{\beta_i\}; \{s_i\}_{i=1}^K) = L_{pred}(\{\beta_i\}; \{s_i\}_{i=1}^K) + \frac{\lambda}{K} \sum_{i=1}^K |\sigma(\beta_i)|; \tag{6}$$

where λ is a hyper-parameter that weights the sparsity constraint against the prediction loss L_{pred}.

4. Residual Coding

For residual coding, we adopt HEVC Test Model (HM-16.7). Specifically, the motion-compensated residuals are compressed in *intra* mode with quantization parameter (QP) adaptation. To this end, the residual frames, having a dynamic range of [-255,255], are uniformly quantized and converted (in a lossy way) into signals of value from 0 to 255 for 8-bit coding. The maximum Coding Unit size is set to 64x64. Remarkably, the compression quality is adjusted by varying the QP value so that every reconstructed video frame has an MS-SSIM value larger than a pre-defined threshold while meeting the bit rate constraint.

5. Experiments

5.1. Settings and Implementation Details

In terms of the number K of critical pixels to retain, we experiment with three settings. The first two set K to 91 (Setting 1) and 282 (Setting 2), respectively, without critical pixel dropout. They correspond roughly to sending 1 to 3 motion vectors per Coding Unit of size 64x64. The third (Setting 3) invokes critical pixel dropout by setting K to 282 and keeping only 91 of them with the largest α_i.

For carrying out the steepest descent update, the values of $\{s_i\}_{i=1}^K$ are initialized to be on a uniform, rectangular 2-D grid that spans across the entire video frame, with their β_i starting at 0. To speed-up the process, the loss $L(\{\beta_i\}; \{s_i\}_{i=1}^K)$ is evaluated on 2x down-sampled I_r, I_t, and F. The resulting $\{s_i\}$ are then scaled to full resolution for motion compensation and residual generation.

As for the hyper-parameters, the number of iterations for parameter update is fixed at 200. The temperature parameter of the sigmoid function is initialized to 5.5 and increased incrementally by 0.25 after each gradient update. The hyper-parameter λ in Eq. (6) is 1e-5.

5.2. Quantitative Comparison

Table 1 present the PSNR and MS-SSIM results over the CLIC validation sequences. We compare two categories of methods: motion-compensated inter-frame prediction (1) *without* and (2) *with* residual coding. Among the methods without residual coding, our POBMC schemes with sparse motion show consistently higher PSNR and MS-SSIM. They all surpass the flow-based motion compensation, which requires sending a dense flow map. This is because POBMC is a multi-hypothesis prediction scheme, which has been proven superior to single-hypothesis prediction. Interestingly, the one with critical pixel dropout (POBMC 282-91) performs comparably to POBMC 282, which uses 3 times more motion vectors, and better than POBMC 91, which uses the same number of motion vectors. As such, POBMC 282-91 + RC achieves the best