Presented at the DLSU Research Congress 2019
De La Salle University, Manila, Philippines
June 19 to 21, 2019

Manifold Intra-Prediction for Image and Video Coding

Carlo Noel Ochotorena[1,2,*] and Yukihiko Yamashita[2]
[1] *De La Salle University, Manila, Philippines*
[2] *Tokyo Institute of Technology, Tokyo, Japan*
**Corresponding Author: carlo.ochotorena@dlsu.edu.ph*

Abstract: Image and video coding is an important aspect of modern imaging – allowing for the use of higher-resolution imaging systems without severely affecting the storage and transmission requirements of the image or video data. Part of most modern coding systems is the use of intra-prediction techniques to reduce the amount of data to be encoded. The current image and video standards describe handcrafted intra-prediction schemes that may not fully exploit the correlation of natural images. This work improves on conventional intra-prediction using a sparsity-regularised regression scheme derived from training images. In addition, it introduces a manifold-adaptive intra-prediction scheme that further reduces the coding residuals obtained from intra-prediction.

Key Words: intra-prediction, image coding, video coding

1. INTRODUCTION

Image and video coding techniques play a mostly unrecognised, yet, integral role in the modern world. With the abundance of cameras and other imaging devices today, accompanied by an increase in the resolution of these devices, the amount of visual information available in the digital realm is continually on the rise. If transmitted in an unprocessed form, this data can easily consume the bandwidth of a transmission line.

Consider, for instance, a video stream operating at the 1080p/60fps standard. Each frame of this stream contains approximately 2 million pixels, each encoded with three colour intensities (red, green, and blue) amounting to about 50 Megabits of information. Over the course of one second, this is equivalent to about 3 Gigabits of data (3 Gbps). Most *local* networks connections only operate at 1 Gbps, while the average global internet speed is only at 5.6 Mbps. In either case, transmitting the video stream becomes impossible without additional processing.

A similar challenge would apply to store images and videos. Without any form of processing, visual data will demand a large amount of storage. For this reason, images and videos are often compressed in a process referred to as image or video coding.

Coding techniques exploit the fact that images are generally piecewise-smooth functions (Ran & Farvardin, 1995). Large portions of an image are smooth and, as such, contain only low-frequency information. This property allows frequency-based transforms such as the discrete Fourier transform, discrete cosine transform (DCT), and the discrete wavelet transform (DWT) to compactly represent the same information. The latter two, in particular, have been used in the JPEG (Wallace, 1992) and JPEG2000 (Taubman & Marcellin, 2013) standards, respectively.

While basis transforms such as DCT and DWT help reduce the amount of information needed to store or transmit small patches in an image, additional savings can also be obtained by exploiting the transmission order of patches. Most coding techniques send out image patches in a sequential manner, often from left to right, and then moving from

the top to the bottom. A consequence of this behaviour is that information about previous patches is already available when encoding and decoding a given patch. Many coding algorithms use this information to guess the content of the current patch, a process known as intra-prediction, to achieve additional savings.

In the JPEG standard, only the most rudimentary prediction is employed, where the current patch is estimated to be a flat image whose intensity is the *average* intensity of all the pixels from the previous patch. Newer video coding standards such as AVC (Sullivan & Wiegand, 2005) and the more modern HEVC (Lainema, Bossen, Han, Min, & Ugur, 2012) utilise more sophisticated intra-prediction techniques based on directional propagation of pixel values or using first-order interpolation, resulting in increased coding efficiency.

The intra-prediction techniques used by AVC and HEVC are handcrafted predictors and, while shown to be effective, may not fully exploit the properties of natural images. An alternative approach is to use a training methodology to design the predictors. For instance, one approach is to use a Markov model to adapt the linear predictors depending on the estimated directionality of the current patch (Garcia & Queiroz, 2010). Another approach is to simply replace the directional predictors of AVC/HEVC with trained regressors (Zhang, Zhao, Ma, Wang, & Gao, 2011).

Our work introduces a similar approach to the latter. However, unlike the existing approach, our technique introduces an iterative approach to training regressors by constantly regrouping training patches based on the optimal regressor. Also, in contrast to the existing works that use a least-squares error criterion for deriving regressors, our approach uses sparsity-regularised optimisation in the DCT domain to further improve coding performance.

Another important contribution in this work is the notion of manifold intra-prediction. Unlike previous approaches, the proposed technique defines a manifold for the current patch based on the structure of neighbouring patches that have already been transmitted. Each manifold then defines a unique set of regressors that more effectively captures the characteristics of the manifold. We show, in this work, that the said approach improves the sparsity of the DCT coefficients which can lend towards more efficient coding in images and videos.

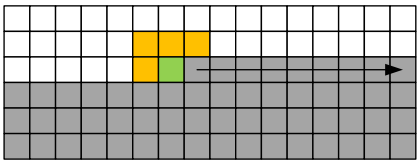

Fig. 1. Transmission of patches in an image. Patches are transmitted from left to right, starting with the topmost row and moving downwards. The white blocks represent patches that have been transmitted while the grey blocks have not yet been transmitted. For the current patch (green), neighbouring patches (orange) can be used for intra-prediction.

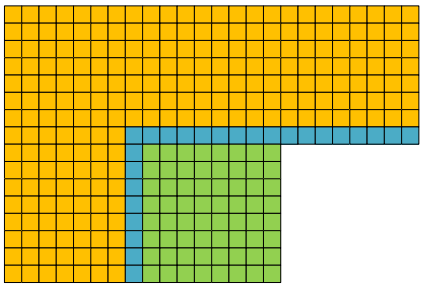

Fig. 2. Pixel-level view of intra-prediction. Conventional intra-prediction uses the blue pixels to predict the values of the green pixels. Our proposed approach employs both blue and orange pixels to determine the manifold of the patch.

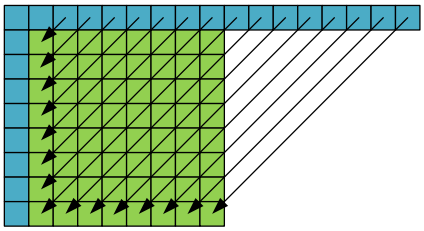

Fig. 3. Conventional directional intra-prediction. The values of the blue pixels are copied following the prediction path.

2. Intra-Prediction

2.1 Conventional Intra-Prediction

Intra-prediction is already embedded in many existing coding standards. The H.264/AVC coding standard describes up to 9 prediction modes, depending on the size of the coding block. The newer H.265/HEVC standard extends this to up to 35

prediction modes, again, depending heavily on the context. In both cases, intra-prediction is based on previously-transmitted patches in the image.

A simplified view (not considering coding tree sub-partitioning) of this can be seen in Fig. 1. For a given patch in the transmission sequence, information regarding its neighbours is usually available from the previous patches to its top and left. Such information can then be used to simply propagate pixel values (see Fig. 2 and Fig. 3) to estimate the pixel values of the new block. Since there are several prediction modes available, the encoder must specify which mode is being used by transmitting additional bits. Often, this additional overhead can be coded at a much lower cost compared to the bit savings incurred by intra-prediction.

2.2 Regression-Based Intra-Prediction

In conventional intra-prediction, each pixel in the current block can be estimated based on one or two pixels from the reference pixels (i.e. the blue region in Fig. 3), depending on the angle described by the intra-prediction mode. If the angle points to an area between two pixels, linear interpolation is used to achieve sub-pixel estimation.

Our proposed regression-based intra-prediction extends this model to make use of all available reference pixels to achieve estimation. If we put together all the reference pixel values in patch i into a vector $\boldsymbol{x}_i$, we can generate the estimate $\boldsymbol{y}_{i,j}$ by multiplying with the j-th regressor $\boldsymbol{R}_j$ (in matrix form):

$$\boldsymbol{y}_{i,j} = \boldsymbol{R}_j \boldsymbol{x}_i \qquad \text{(Eq. 1)}$$

Intuitively, the best prediction mode k_i is the mode that minimizes the prediction error:

$$k_i = \underset{j}{\operatorname{argmin}} \left\| \boldsymbol{y}_{i,j} - \boldsymbol{R}_j \boldsymbol{x}_i \right\|_2^2 \qquad \text{(Eq. 2)}$$

The problem with the Eq. 2 is that minimisation of prediction errors does not necessarily improve the coding performance. The resulting error may introduce frequency components which need additional coding. A more appropriate criteria, used in this work, would be to directly evaluate the sparsity of the error in the DCT domain:

$$k_i = \underset{j}{\operatorname{argmin}} \left\| \boldsymbol{D}\left(\boldsymbol{y}_{i,j} - \boldsymbol{R}_j \boldsymbol{x}_i\right) \right\|_1 \qquad \text{(Eq. 3)}$$

where the matrix $\boldsymbol{D}$ describes the fixed DCT transform matrix. Since there are only a few prediction modes, the solution can be found by calculating the associated error with each mode.

To train the regressors, random sample patches are collected from all 24 images of the Kodak image database (Franzen, 2002). A total of 25000 patches are randomly obtained from each training image. To promote meaningful regressors, smooth patches (based on a norm-criterion) are rejected from the random selection process. In addition to this, in order to avoid rotation bias in the training patches, each patch is flipped and rotated in 90° increments to form a total of eight (8) transformations for each patch. This effectively results in 200000 training patches obtained from each image and 4.8 million patches from the entire database.

Given these training patches, prediction is carried out using the original 35 HEVC intra-prediction modes. Once the optimal prediction mode for each training patch is found, we can gather all the patches utilising a given prediction mode:

$$(X_k, Y_k) = \left\{ (x_i, y_i) \mid k = \underset{j}{\operatorname{argmin}} \left\| \boldsymbol{D}\left(\boldsymbol{y}_{i,j} - \boldsymbol{R}_j \boldsymbol{x}_i\right) \right\|_1 \right\} \text{(Eq. 4)}$$

Using the collated patches, we can then define an optimisation problem to update the regressor as follows:

$$\boldsymbol{R}_k = \underset{\boldsymbol{R}}{\operatorname{argmin}} \|\boldsymbol{D}(\boldsymbol{Y}_k - \boldsymbol{R}\boldsymbol{X}_k)\|_1 \qquad \text{(Eq. 5)}$$

This problem does not have a closed form solution so, instead, we approach this problem using an augmented Lagrangian technique (Boyd, Parikh, Chu, Peleato, & Eckstein, 2011):

$$\boldsymbol{R}_k = \underset{\boldsymbol{R}}{\operatorname{argmin}} \|\boldsymbol{E}\|_1 + \frac{\beta}{2} \|\boldsymbol{E} - \boldsymbol{D}(\boldsymbol{Y}_k - \boldsymbol{R}\boldsymbol{X}_k) - \boldsymbol{\Gamma}\|_2^2 \text{ (Eq. 6)}$$

where $\boldsymbol{E}$ is an auxiliary variable used in the optimisation process and $\boldsymbol{\Gamma}$ represents the Lagrangian multiplier (in additive form after augmentation). The optimisation problem described in Eq. 6 can be solved over several iterations. In our specific implementation, we utilise 5 iterations.

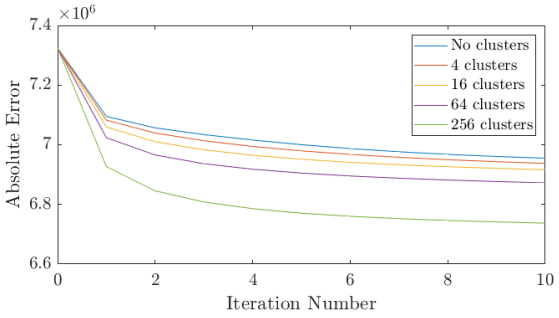

Fig. 4. Prediction errors with Manifold Intra-Prediction. Note the decreasing errors associated with larger clusters.

2.3 Manifold Intra-Prediction

Another technique we introduce to further improve coding efficiency is to assume that patches behave differently within their respective manifolds. As a result of this assumption, patches can be clustered using the available pixels (orange and blue pixels in Fig. 2) and subsequently, regressors are refined within each cluster. Clustering is performed using a cosine-distance criterion which is equivalent to clustering normalised patches on the surface or a hypersphere.

3. RESULTS AND DISCUSSION

3.1 Prediction Errors

In order to evaluate the proposed technique, we train regressors for varying cluster sizes. Each set of regressors is then tested on random patches to evaluate the sparsity of the DCT errors. The results of this test can be seen in Fig. 4. A lower error value in this test indicates that more of the DCT errors are equal to zero which is useful in coding. It should also be noted that when no clusters are used, the system is equivalent to using regression-based intra-prediction with no manifolds.

Before any regressor refinement is performed, (iteration 0), the regressors in each test are identical to the HEVC intra-predictors. This allows us to establish a baseline for the performance of the intra-prediction scheme. After the first iteration, we note that all variants of the proposed technique show a substantial decrease in the DCT errors. Subsequent refinements to the regressors result to further improvements in the residual magnitude, but the amount of reduction seen is less drastic beyond the first iteration. In our experiments, we limit the number of iterations to ten (10) as the reductions beyond this no longer warrant the additional computation time.

Throughout all the iterations, it is easy to see in Fig. 4 that all variants of our proposed method improve on the perform better than the HEVC baseline. More importantly, an increase in the number of clusters *consistently* corresponds to a reduction in the DCT residuals. This demonstrates that dividing patches into manifolds allows the regressors to better capture the relationship between the reference pixels and the pixels to be estimated.

3.2 Complexity

Compared to the original HEVC intra-prediction scheme, the proposed method adds a few processes to the encoding and decoding scheme and would naturally affect their performance. It is, therefore, important to inspect the complexity of the proposed method.

Perhaps the most significant addition to the encoding and decoding pipeline introduced by our proposed technique is the clustering of patches into their respective manifolds. For an encoder working with m clusters on $n \times n$ blocks, the resulting complexity at the block level is $\mathcal{O}(mn^2)$. If we consider that an image has p pixels in total, this would result in approximately p/n^2 blocks. This means that for any given image, the overall complexity of the clustering process is $\mathcal{O}(mp)$, indicating that the system remains linear with respect to the number of input pixels.

The second change relative to HEVC is the use of more complex regression operators. It should be noted, however, that training is performed offline—once the refined regressors have been found, they can be used for any new images or videos fed to the encoder, thus avoiding the iterative optimisation during the encoding and decoding process. This is similar to how HEVC uses a set of *fixed* predictors to generate pixel estimates. However, instead of using two (2) to four (4) pixels as inputs to the prediction

process, our proposed method uses all $3n + 1$ reference pixels (see Fig. 3). This leads to a block-level complexity of $\mathcal{O}(n^3)$ or an image-level complexity of $\mathcal{O}(np)$. Like the clustering process, this is, again, linear with respect to the number of input pixels. Overall, the proposed method does increase the computational time needed to achieve intra-prediction but maintains the linear complexity of HEVC with respect to the input size.

4. CONCLUSIONS

In this work, we introduced several techniques to construct regressors to perform intra-prediction on images and videos. Our proposed approach uses direct sparsity-based optimisation to construct improved intra-predictors that are capable of providing good estimates of a patch based on previous patches. The use of our proposed manifold intra-prediction scheme further enhances the intra-prediction performance.

It should be pointed out that the above results operate on isolated testing of the intra-prediction system. In reality, such a system is only a part of a more complex coding scheme. For this reason, it is also useful to investigate the effects of the proposed scheme as part of a complete system. Future work must be done to test and validate the proposed scheme in a fully functional coder such as HEVC.

5. REFERENCES

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning archive, 3*(1), 1-122.

Garcia, D., & Queiroz, R. (2010). Least-Squares Directional Intra Prediction in H.264/AVC. *IEEE Signal Processing Letters, 17*(10), 831-834.

Lainema, J., Bossen, F., Han, W.-J., Min, J., & Ugur, K. (2012). Intra Coding of the HEVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology, 22*(12), 1792-1801.

Ran, X., & Farvardin, N. (1995). A perceptually motivated three-component image model-Part I: description of the model. *IEEE Transactions on Image Processing, 4*(4), 401-415.

Sullivan, G., & Wiegand, T. (2005). Video Compression - From Concepts to the H.264/AVC Standard. *Proceedings of the IEEE, 93*(1), 18-31.

Taubman, D., & Marcellin, M. (2013). JPEG2000 Image Compression Fundamentals, Standards and Practice. *Journal of Electronic Imaging, 11*(2), 286-287.

Wallace, G. (1992). The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics, 38*(1).

Zhang, L., Zhao, X., Ma, S., Wang, Q., & Gao, W. (2011). Novel intra prediction via position-dependent filtering. *Journal of Visual Communication and Image Representation, 22*(8), 687-696.