K-NN SEARCH USING LOCAL LEARNING BASED ON REGRESSION FOR NEIGHBOR EMBEDDING-BASED IMAGE PREDICTION

Christine GUILLEMOT, Safa CHERIGUI**, Dominique THOREAU***

(*): INRIA, Campus Universitaire de Beaulieu, 35042 RENNES, FRANCE
(**): Technicolor R & D France, 1, rue du Clos Courtel, 35576 Cesson-Sévigné, FRANCE
Contact author: Christine.Guillemot@inria.fr

ABSTRACT

The paper describes a K-NN search method aided by local learning of subspace mappings for the problem of neighbor-embedding based image Intra prediction. The local learning of subspace mappings relies on multivariate linear regression. The method is used jointly with Locally Linear Embedding (LLE) as well as with a method inspired from Non Local Means (NLM) for prediction. Linear and kernel ridge regression are also considered directly for predicting the unknown pixels. Rate-distortion performances are then given in comparison with Intra prediction using LLE and classical K-NN search, as well as in comparison with H.264 Intra prediction modes.

Index Terms— Image compression, prediction, linear regression, data dimensionality reduction

1. INTRODUCTION

Intra prediction is a key component of image and video compression algorithms. Given observations, or known samples in a spatial neighborhood, the goal is to estimate unknown samples of the block to be predicted. For example, in H.264/AVC, there are three Intra-frame prediction types called Intra-16x16, Intra-8x8 and Intra-4x4 [1]. Each block is predicted from prior encoded pixels of spatially neighboring blocks. In the Intra-4x4 prediction mode, in addition to the "DC" mode which predicts the entire 4x4 block from the mean of neighboring pixels, eight directional prediction modes have been specified. The prediction is done by simply "propagating" the pixel values along specified directions. This approach is suitable in the presence of contours when the directional mode chosen corresponds to the orientation of the contour. However, it fails in more complex textured areas.

An alternative spatial prediction algorithm based on template matching (TM) has been described in [2]. A template is formed by previously encoded pixels in a close neighborhood of the block to be predicted. The best match between the template of the block to be predicted and candidate texture patches of same shape, within a causal search window, allows finding the predictor of the block to be predicted. In [3], we have considered neighbor embedding (NE) solutions (e.g. LLE [4] and NMF [5]) to address the problem. Intra prediction methods using NE first search, within a window in the causal part of the image, for the K-nearest neighbors to the template pixels of the input block to be predicted. They then search for the best approximation of the template pixels by a linear combination of their K-NN. The methods then vary in the way the coefficients of the linear combinations are computed. The NLM-inspired method computes similarity weights with a Gaussian kernel as in [6] for denoising or in [7] for inpainting. In the LLE-based and NMF-based approaches, the weights are given by least squares approximations under a sum-to-one constraint (LLE), and non-negativity constraint (NMF). Significant gains have been shown in [3] when comparing the NE-based intra prediction against a simple TM.

However, the K-NN patches used for the linear approximation of the input patch has obviously a strong impact on the performance. Searching for the K-NN by computing a distance on the template pixels may not lead to the best blocks for approximating the unknown pixels of the block to be predicted, especially in the case where there are discontinuities between the template and the current block. In order to have K-NN patches which are similar in terms of distance computed on the template, while being also relevant for the block to be predicted, the idea is to use mapping functions between subspaces where lie the known and unknown parts of the input patch. These mapping functions are learned from training patches within the known part of the image, using multivariate linear regression. These mapping functions are then used for computing a first estimate of the unknown pixels. This first estimate is then used to help the K-NN search, so that the K-NN blocks considered in the neighbor embedding are relevant for the block to be predicted and not only to its template. Linear as well as kernel ridge regression are also considered directly for prediction. The performances of the enhanced K-NN search are assessed in the context of two neighbor embedding solutions, using similarity and LLE [4] weights. Simulation results show the very good performances of the regression based methods as well as the gain in terms of PSNR versus rate when using the K-NN search aided by local learning of linear subpace mappings.

The rest of the paper is organized as follows. Section 2 reviews the image prediction problem. Section 3 reviews the neighor embedding methods considered in this paper. Section 4 describes the proposed enhanced K-NN search algorithm using subspace mappings learned with multivariate linear regression. Section 5 briefly recalls the compression algorithm used to assess the performances of the methods and then gives prediction and compression performance illustrations.

2. THE PREDICTION PROBLEM

Let X be a texture block which comprises a known part X^k (of a given shape) formed by the pixels located in a causal neighborhood and an unknown part X^u formed by the block to be predicted (see Fig.1). For each input texture block X, we constitute a training set of patches by taking all blocks in a search window SW within the coded-decoded causal part of the image. Each block of the training set is also formed by a so-called "*known*" part (set of N_1 pixels at the same positions as the known pixels X^k of X, also referred to as the template) and an "*unknown*" part (set of N_2 pixels at the same positions as the unknown pixels X^u of X). We assume that the set of data points formed by the *known* template pixels and the set of data points formed by the complementary (or *unknown*) parts in input and training patches belong to two related manifolds. The goal of the proposed methods is to "*connect*" these two manifolds, and this in order to make sure that a good approximation of the known part of the input block (i.e. of the template) will also lead to a good approximation of the block to be predicted.

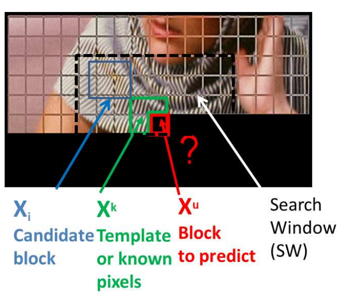


Fig. 1. Problem statement and notations: X^k is the approximation support (known pixels), X^u is the block to be predicted, and SW is the search window from which training blocks are taken.

3. NEIGHBOR EMBEDDING: A REVIEW

The TM algorithm searches for the best match between the template $\mathbf{X}^k$ and possible candidate blocks $\mathbf{X}_i^k$ (of the same shape as the template) taken from the causal search window. The template matching technique searches for the candidate block which minimizes the distance d_i as $\text{argmin}_{i\in\{1...M\}}\{d_i\}$ where $d_i = ||X^k - X_i^k||_2^2,\ \ i = 1...M$, where M is the number of all possible candidate blocks in the search window. The predicted block $\tilde{X}^u$ is then simply obtained by copying the pixel values of the candidate X_i^u minimizing the above distance.

Non Local Mean (NLM): Instead of retaining the block which is the most similar to the template pixels, one can keep the K blocks most similar to the template pixels. These K-NN can then be linearly combined to find an approximation of the block to be predicted. This idea has been used in [7] for inpainting, where the authors compute the weights with a similarity kernel function in order to give higher weights to patches which are more similar to the known samples of the input patch to be inpainted. This approach, proposed for inpainting in [7] has been inspired from the non local means (NLM) algorithm used for de-noising in [8] and for texture synthesis in [6]. After obtaining K-NN blocks $X_i, i = 1$...K,, the prediction follows by $\tilde{X}^u = \sum_{i=1}^{K} \alpha_i X_i^u$, where

$$\alpha_i = \exp\left(-\frac{\left\|X^k - X_i^k\right\|_2^2}{h}\right) \tag{1}$$

and h is a decay coefficient. Note that the calculated weights are finally normalized, i.e., $\alpha_i = \alpha_i / \sum_i \alpha_i$.

LLE: Similarity weights do not lead to the best approximation of the input block. One can instead used least squares approximation, possibly using different constraints on the weights of the linear combination. The goal is thus to best approximate the input block from its K nearest neighbors, hence the name of neighbor embedding. Once the K nearest neighbors (K-NN) to the input block are found, using the LLE method [4], the neighbor embedding is formulated as a least square approximation of the input data vector (or here block) under the constraint that the weights of the linear approximation sum to one, i.e., as

$$\text{argmin } E(W) = \left\|X^k - \sum_i w_i X_i^k\right\|_2^2 \text{ s.t. } \sum_i w_i = 1; \tag{2}$$

Each weight w_i can be computed as $w_i = \frac{y_i}{\sum_i y_i}$ where y_i is solution of the linear system $(D^T D)y = \mathbf{1_K}$. The term D denotes the $N_1 \times K$ neighborhood matrix of the vector formed by the N_1 known pixels X^k of the input patch X. The i^{th} column of the matrix D is $X_i^k - X^k$, where X_i^k is the i^{th} neighbor of X^k. The notation $\mathbf{1_K}$ stands for the column vector of ones of dimension K. In practice, the linear system of equations $(D^T D)y = \mathbf{1_K}$ is solved, and then the weights are rescaled so that they sum to one.

4. K-NN SEARCH USING LOCAL LEARNING OF SUBSPACE MAPPINGS

All the neighbor embedding methods described above require first searching for the K-NN to the input block to be predicted. These K-NN are searched by computing a Euclidean distance between the template pixels (the known pixels) of the input block and the pixels at the same positions in all the candidate blocks X_i taken from the search window SW. The goal, when using neighbor embedding methods in the prediction problem, is to approximate each input "template" block (which can be seen as a data point in a space of dimension N_1) by a linear combination of its K-NN. Once the weights of the linear combination are found, the same weights are kept for combining the complementary part formed by pixels at the same position as the unknown pixels, and this sub-block can be seen as a data point in a space of dimension N_2. This principle is illustrated in Fig. 2. However, the K-NN found to provide a good approximation of the template pixels may not be good candidates for approximating the unknown pixels of the block to be predicted. In order to enhance this K-NN search for the prediction problem, we consider here a local learning of the correspondence between the template (or so-called *known part*) and the corresponding *"unknown part"* of all the candidate (or training) blocks X_i located within the search window SW in the neighborhood of the block to be predicted. This correspondence is represented as a mapping function F_1 (as shown in Fig. 2) which can be learned using regression methods.

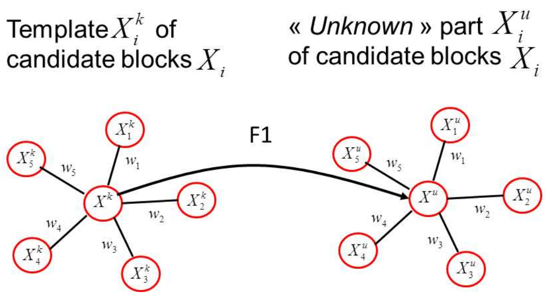


Fig. 2. Learning the mapping function between *known* parts $X_i^k \in \mathbf{R^{N_1}}$ and corresponding adjacent subblocks $X_i^u \in \mathbf{R^{N_2}}$ of all candidate blocks taken from the search window.

The first method considered for learning the mapping function F_1 is a multivariate linear regression. The problem is therefore of searching for the function F_1 minimizing $||(M_u)^T - (M_k)^T F_1^T||^2$ which is of the form $||Y - XB||^2$ (corresponding to the linear regression model $Y = XB + E$), the minimization of which (by zeroing the derivative with respect to W) gives the least squares estimator

$$F_1 = M_u M_k^T (M_k M_k^T)^{-1} \tag{3}$$

where M, M_u and M_k are the matrices whose columns are formed by all the candidate blocks and their *unknown* X_i^u and *known* parts X_i^k, $i = 1 \ldots N$, respectively, with N depending on the size of the search window.

The method then proceeds as follows. A first estimate of the block to be predicted is computed as

$$X^k \xrightarrow{F_1} \tilde{X}^u. \tag{4}$$

We then search for the K-NN $X_i, i = 1 \ldots K$ of the block $\tilde{X}$ formed by both its template X^k and the first estimate $\tilde{X}^u$ of the unknown pixels of the input block, i.e., of

$$\tilde{X} = \begin{bmatrix} X^k \\ \tilde{X}^u \end{bmatrix}.$$

The estimate $\tilde{X}^u$ gives a prediction of the unknown block using a linear regression learned from the block local neighborhood. This prediction will be referred to as **LR** in the experimental section and curves.

The algorithm then proceeds to the neighbor embedding of the vector $\tilde{X}$ by searching for its linear approximation from its K-NN. The weights of the linear approximation can be computed using LLE (Eqn.(2)), NLM (Eqn.(1)), or any other neighbor embedding technique.

Note that kernel regression, using a Gaussian kernel, has also been considered instead of linear regression for learning the mapping function. However, the gain obtained when using the kernel regression rather than the linear regression is small compared to the extra complexity induced. This gain is however illustrated in Fig. 4 when using the learned mapping function F_1 directly for prediction.

5. EXPERIMENTAL RESULTS

Fig. 3. Test images.

The prediction methods are compared with H.264 Intra directional prediction modes, however, with a simple residue encoder as described in [9]. The test images used are shown in Fig. 3. The top 4 rows and left 4 columns of blocks of size 4x4 are predicted with the H.264/AVC Intra prediction modes. The algorithm then proceeds with the prediction based on NE methods, using 7 template shapes. Once a block has been predicted, the DCT transformed residue is quantized with a uniform quantizer and encoded with an algorithm similar to JPEG. A uniform quantization matrix with $\Delta = 16$ is weighted by a value $\mathbf{w}_{q_f}$ which depends on a quality factor q_f (varying from 10 to 90 with a step size of 10) as $\mathbf{w}_{q_f} = 50/q_f$ if $q_f \leq 50$ and $\mathbf{w}_{q_f} = 2 - 0.02q_f$ if $q_f > 50$. Image blocks are processed in a raster scan order, and the reconstructed image is obtained by adding the quantized residue to the prediction. The training patches are collected from the search region

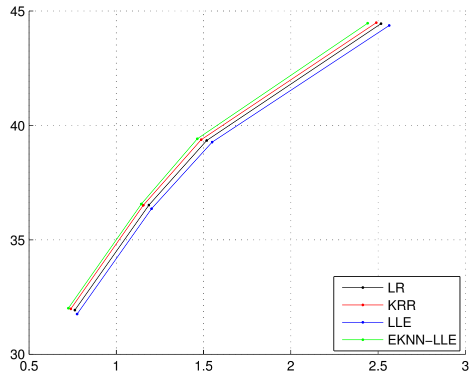


Fig. 4. Rate-distortion (RD) performances for Cameraman with LR and KRR, LLE with classical K-NN search on template, and with K-NN search aided by local learning(EKNN-LLE).

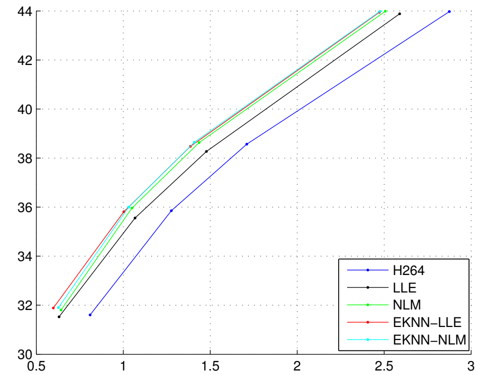
which is located in a causal neighborhood of the block to be predicted. The parameter K is set to 10 in the experiments.

5.1. Kernel ridge versus linear regression for prediction

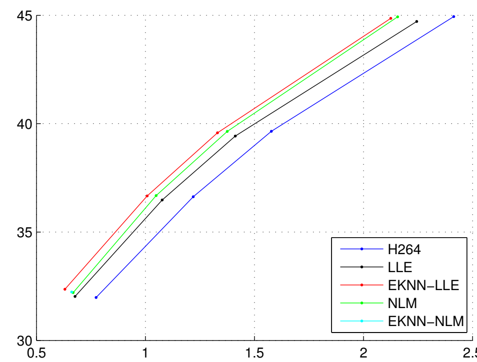
Fig.4 shows the PSNR versus rate performances obtained for the Cameraman image when using linear (LR) and kernel ridge regression (KRR) for prediction. A small gain can be observed when using kernel ridge regression, however at the expense of higher complexity. In the sequel, only linear regression has been used.

5.2. Linear-regression based K-NN search

Fig.5 shows the PSNR versus rate performances obtained when using the enhanced K-NN search followed by the LLE-based neighbor embedding, or by NLM, in comparison with LLE and NLM with K-NN searched on the template only, and in comparison with H.264, for the images Barbara and Roof. The simulation results show a significant gain when using the enhanced K-NN search with LLE. The plots also show good performances when using the NLM-based method, which is below the ones obtained with LLE for the Roof image but which are very close to those of LLE for the Barbara image. A significant improvement has also been observed when using the enhanced K-NN search with other neighbor embedding techniques like LLE-LDNR [10] or NMF [5]. However, the K-NN search aided by the regression-based subspace mapping brings only a little improvement when the neighbor embedding which follows uses NLM. This is due to the fact that similarity weights in NLM-based prediction does not search to approximate the input vector, unlike in methods like LLE which perform a constrained least square approximation of the input vector.



Barbara



Roof

Fig. 5. Rate-distortion (RD) performances for Barbara (top) and Cameraman (bottom), with LLE and NLM, with classical K-NN search on template and with improved K-NN search using local learning.

6. CONCLUSION

This paper has introduced an enhanced K-NN search method for image Intra prediction based on neighbor embedding methods. The K-NN search is aided by a first estimate done via mapping functions between subspaces formed by the templates and their complementary part in the input and training blocks. These mapping functions are learned locally using regression. The use of these functions has also been considered as a prediction method, without further refinement with NE-based approximation. Substantial gains have been shown when compared to existing NE-based prediction techniques. Note finally that the method naturally applies to other problems such as inpainting or loss concealment.

7. REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE*

Trans. on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[2] T. K. Tan, C. S. Boon, and Y. Suzuki, “Intra prediction by template matching,” in *Proc. IEEE Int. Conf. Image Process.*, 2006, pp. 1693–1696.

[3] M. Turkan and C. Guillemot, “Image prediction based on neighbor embedding methods,” *IEEE Trans. on Image Processing*, vol. 21, no. 4, pp. 1885–1898, Apr. 2012.

[4] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, pp. 2323–2326, Dec. 2000.

[5] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” *Advances in Neural Information Process. Syst. (NIPS)*, 2000.

[6] A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *Proc. IEEE Int. Conf. Computer Vis.*, vol. 2, 1999, pp. 1033–1038.

[7] A. Wong and J. Orchard, “A nonlocal-means approach to exemplar-based inpainting,” in *IEEE Int. Conf. Image Process. (ICIP)*, 2006, pp. 2600–2603.

[8] A. Buades, B. Coll, and J. M. Morel, “A review of image denoising algorithms, with a new one,” *SIAM J. Mult. Scale Modeling Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

[9] M. Turkan and C. Guillemot, “Image prediction: Template matching vs. Sparse approximation,” in *IEEE Int. Conf. Image Process. (ICIP)*, 2010, pp. 789–792.

[10] Y. Goldberg and Y. Ritov, “Ldr-lle: Lle with low-dimensional neighborhood representation,” in *Proceedings of the 4th International Symposium on Advances in Visual Computing, Part II*, ser. ISVC ’08, 2008, pp. 43–54.