# Locally linear embedding based texture synthesis for image prediction and error concealment

Mehmet Turkan, Christine Guillemot

# LOCALLY LINEAR EMBEDDING BASED TEXTURE SYNTHESIS FOR IMAGE PREDICTION AND ERROR CONCEALMENT

*Mehmet Türkan and Christine Guillemot*

INRIA
Campus Universitaire de Beaulieu, 35042 Rennes, France
firstName.lastName@inria.fr

**ABSTRACT**

The template matching algorithm is a simple extension to exemplar-based texture synthesis. Average of template matching predictors or non-local means based approaches can be seen as heuristic extensions to template matching. These methods which linearly combine several texture patches have been shown to be more robust in synthesis and to give better results when compared to simple template matching. However, they do not search to minimize an approximation error on the known pixel values in the template. They are rather heuristic methods for calculating the linear weighting coefficients. This paper proposes a neighbor embedding based texture synthesis method by formulating the problem as a least-squares optimization using locally linear embedding. By this means, one calculates the linear weighting coefficients by solving a constrained optimization for approximating the template. The proposed texture synthesis framework has first been applied to the image prediction (predictive coding) problem. It has then been extended to a loss concealment application for transmission errors. Experimental results demonstrate the effectiveness of the proposed method for both image compression and error concealment.

***Index Terms*—** Template matching, average template matching, non-local means, locally linear embedding, image prediction, error concealment

## 1. INTRODUCTION

Texture synthesis is a key component of various image and video processing applications such as inpainting, restoration, (predictive) coding, and so on. Existing texture synthesis methods can be classified into two main categories. The first category relies on either the use of partial differential equations [1] or variational approaches [2]. The ideas presented in this type basically try to propagate via *diffusion* both geometric and photometric information that hits the border of the region to be synthesized. These methods are known to work relatively well for synthesizing small regions in an image. However, they tend to introduce blur for large regions, and the results are highly dependent on the input parameters. The second category of texture synthesis concerns exemplar-based methods [3, 4, 5] which sample texture patches (or pixels) from an other image (or from the image itself) in order to synthesize new textures. Exemplar-based methods are known to work well in cases of regular textures, and they have widely been used in a large variety of applications because of their simplicity and efficiency. Moreover, these methods can compute a priority for the different patches to be synthesized so that the algorithm can start by first propagating important structures in the image. For example, the authors in [6] introduce an exemplar-based method in which a filling order is defined by a priority function which depends on the angle between the isophote direction and the normal direction of the local filling front. The goal of this priority function is to ensure that the linear structures are propagated before texture filling.

Exemplar-based methods usually copy a single patch (or a pixel) from an input texture sample (*exemplar*) to the output texture according to the known local neighboring set of pixels of the current patch (or pixel) to be synthesized. The well-known *template matching* (TM) algorithm operates as an extension to exemplar-based texture synthesis, and has been shown to be an effective method for several applications including intra and inter prediction [7, 8], and inpainting [6, 9]. A so-called *template* is formed by the known pixels in a close neighborhood of the unknown patch to be synthesized. The best match between the template and the candidate texture patch neighborhood (of the same shape as template), either within a search window or in the whole image, allows finding the predictor of the unknown pixel values.

One can even combine several image patches instead of using a single "best" patch. The simplest way of combining several patches is to assign uniform weights for each patch. This method is the same with the average of multiple template matching (ATM) predictors as proposed in [10]. Recently, non-local means (NLM) based approaches [11, 12] have also been proposed for different texture synthesis problems. These methods which combine several patches have been shown to be more robust in estimating the unknown values and producing better synthesis results. However, the approaches ATM and NLM do not search to minimize an approximation error on the template signal. They are rather heuristic methods to calculate the linear weighting coefficients.

In this paper, we propose a texture synthesis method based on the locally linear embedding (LLE) technique [13]. The underlying motivation can be thought of as finding a best linear combination of a number of "texture primitives" (texture patches) to approximate the input patch using a neighbor embedding optimization rather than a heuristic calculation of weighting coefficients. In LLE, the goal is to reconstruct each input signal from its $K$ nearest neighbors ($K$-NN).

We focus particularly on texture synthesis for two different image processing problems: (i) predictive coding (image prediction) and (ii) error concealment. Although there is a strong similarity between the prediction problem and the error concealment problem, there are at the same time important differences. In concealment, the known samples are not restricted to be in a causal neighborhood and the texture patches can be treated in a different order than the raster scan. However, in prediction, the processing order is the raster scan, the known samples are located only in a causal neighborhood of the block to be predicted. The priority order as used in error concealment can not be used as such (or not with as much freedom)

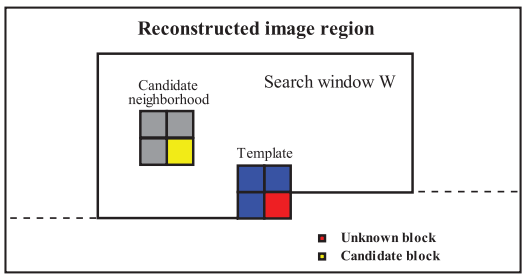


**Fig. 1**. Block-based intra prediction based on template matching. A template is formed by previously encoded pixels in the close neighborhood of the unknown block. The best match between the template and the candidate neighborhood, within the search window W, allows finding the predictor (candidate block) of the unknown block.

for the prediction problem. On the other hand, in an error concealment application, errors in texture synthesis are very critical and can propagate very easily.

The rest of the paper is organized as follows. In Sec. 2 we start with the image prediction problem by revisiting the background approaches including TM, ATM, and NLM. We then introduce the LLE based texture synthesis framework for image prediction. In Sec. 3 we extend the proposed texture synthesis method for an error concealment application. We give then experimental results obtained for image prediction[1] and error concealment in Sec. 4. Finally, Sec. 5 concludes this work with a brief conclusion.

## 2. LOCALLY LINEAR EMBEDDING FOR PREDICTION

### 2.1. Background approaches: TM, ATM, NLM

The main idea of TM consists in estimating the unknown pixel values using image patches in a local search window where all the pixel values are known and available for processing. More precisely, the values of each pixel to be predicted are determined by comparing their spatial neighboring pixels (template) with all candidate neighborhoods in the search window, and assigning the candidate pixel values as a predictor for the unknown sample values by minimizing a distance between the template and the candidate neighborhood. The distance measures used are usually classical ones such as the sum of squared error (SSE). Fig. 1 demonstrates a simple illustration of the TM algorithm for block-based intra image prediction.

Let S denote a region in the image containing the unknown block of size $n \times n$ and its template as shown in Fig. 1. The region S contains 4 blocks hence of size $\mathrm{N} = 2n \times 2n$ pixels for running the TM algorithm. Suppose that the columns $\mathbf{a}_m$ of a matrix $\mathbf{A} \in \mathbb{R}^{\mathrm{N} \times \mathrm{M}}$ are constructed by stacking the luminance values of all possible patches, of size $2n \times 2n$ pixels, in a given causal search window W in the reconstructed image region. Here the matrix $\mathbf{A}$ will be referred to as the *dictionary*. Assume also that the N sample values of the region S are stacked in a column vector $\mathbf{b}$.

For the first step, that is the search for a best match of the known pixel values in the template, the matrix $\mathbf{A}$ is modified by masking its rows corresponding to the spatial location of the pixels of the unknown block. A compacted dictionary $\mathbf{A}_c$ of size $3n^2 \times \mathrm{M}$ is obtained in which actually the columns $\mathbf{a}_{c_m}$ correspond to the possible *candidate neighborhoods* in the search window. The vector $\mathbf{b}$ is also compacted in $\mathbf{b}_c$ with the known pixels in the template of size $3n^2$ values. The TM algorithm then proceeds by calculating

$$m_{opt} = \underset{m \in [1 \ldots \mathrm{M}]}{\arg\min} \{d_m : d_m = \mathrm{DIST}(\mathbf{b}_c, \mathbf{a}_{c_m})\} \tag{1}$$

where the operator DIST denotes a distance metric such as SSE. The prediction signal $\hat{\mathbf{b}}_t$ is simply assigned by the sample values of the candidate block as $\hat{\mathbf{b}}_t = \mathbf{a}_{t_{m_{opt}}}$. The columns $\mathbf{a}_{t_m}$ of $\mathbf{A}_t$ correspond to the possible *candidate blocks* in the search window, where the matrix $\mathbf{A}_t$ is obtained by masking the rows of the dictionary $\mathbf{A}$ corresponding to the spatial location of the pixels of the template.

Starting with the above definition of simple TM, a weighted linear combination of multiple TM predictors can be a possible extension since there might be more than one candidate block which are equally important, or with a varying degree of importance but useful to be used in the prediction process. Let the vector $\hat{\mathbf{b}}_t$ be predicted as a weighted average of the candidate blocks $\mathbf{a}_{t_{m_k}}, k = 1 \ldots K$, where the corresponding candidate neighborhoods $\mathbf{a}_{c_{m_k}}$ taken from the search window are the $K$-NN of the template $\mathbf{b}_c$. A simple consideration ATM is of the weighting coefficients which are all equal to each other. The prediction follows by

$$\hat{\mathbf{b}}_t = \mathbf{A}_t^K \boldsymbol{\alpha} \tag{2}$$

where $\mathbf{A}_t^K$ is the matrix containing $\mathbf{a}_{t_{m_k}}$ in its columns and the elements $\alpha_k$ of the weighting vector $\boldsymbol{\alpha}$ are all equal to $1/K$.

Similar to ATM, the NLM based method tries to aggregate multiple image patches as a weighted linear combination but the weighting coefficients are calculated in a different way. The idea here is to express the weights in terms of the amount of similarity between the candidate neighbor patches and the template, i.e., the contribution weights are calculated with a *patch similarity based kernel function* in order to give more weight to the neighboring patches which are more similar to the template than the others. By using an exponential kernel, the elements $\alpha_k$ of the weighting coefficients vector $\boldsymbol{\alpha}$ can be calculated as $\alpha_k = \exp\left(-\frac{\mathrm{DIST}\left(\mathbf{b}_c, \mathbf{a}_{c_{m_k}}\right)}{h}\right)$ where $h$ is a decay coefficient. The calculated weights are normalized to sum-to-one, i.e., $\boldsymbol{\alpha} = \boldsymbol{\alpha}/sum(\boldsymbol{\alpha})$, in order to avoid a possible overflow in the predicted values of the pixels. Finally, the prediction follows Eqn. 2.

### 2.2. Prediction based on LLE

In this study, we propose to use LLE for calculating the weighting coefficients in the vector $\boldsymbol{\alpha}$. We thus search for an approximation of the template by a linear combination of its $K$-NN, and then keep the same weighting coefficients in the linear combination of the co-located pixels in order to estimate the unknown values of the block to be predicted. In terms of LLE, this optimization can be written as

$$\underset{\boldsymbol{\alpha}}{\arg\min} ||\mathbf{b}_c - \mathbf{A}_c^K \boldsymbol{\alpha}||_2^2 \text{ subject to } \mathbf{1}^{\mathrm{T}} \boldsymbol{\alpha} = 1 \tag{3}$$

and the optimal weighting coefficients in $\boldsymbol{\alpha}$ are computed as

$$\boldsymbol{\alpha} = \frac{\mathbf{G}^{-1}\mathbf{1}}{\mathbf{1}^{\mathrm{T}}\mathbf{G}^{-1}\mathbf{1}}. \tag{4}$$

Here $\mathbf{A}_c^K$ is the matrix containing $\mathbf{a}_{c_{m_k}}, k = 1 \ldots K$, in its columns, $\mathbf{G}$ denotes the Gram matrix of $\mathbf{A}_c^K$ in reference to $\mathbf{b}_c$, and $\mathbf{1}$ is the column vector of ones. In practice, instead of an explicit inversion of the matrix $\mathbf{G}$, the linear system of equations $\mathbf{G}\boldsymbol{\alpha} = \mathbf{1}$ is solved, then the weights are rescaled so that they sum to one. Note that the

[1] This study analyses particularly the performance of ATM, NLM, and LLE for image prediction. A more general performance comparison between TM-based prediction and standard prediction (e.g., H.264/AVC intra) with advantages and disadvantages is available in [14].

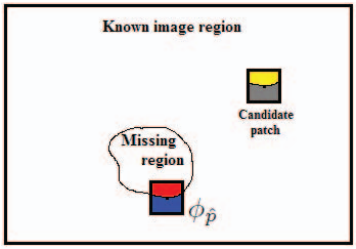


**Fig. 2**. Error concealment. The patch $\phi_{\hat{p}}$ is centered on the highest priority pixel $\hat{p}$ located on the fill front. The template information of $\phi_{\hat{p}}$ (blue) is used for selecting the patches from the known image region and then filling-in the unknown values of $\phi_{\hat{p}}$ (red).

sum-to-one constraint of weights forces the reconstruction of each patch to lie in the subspace spanned by its neighbors. Since the input texture patch samples are all non-negative, the predicted block will also be non-negative. However, the calculated weights can be positive or negative. Finally, the prediction follows Eqn. 2.

## 3. EXTENSION TO ERROR CONCEALMENT

Error concealment is very similar to the image inpainting problem where one can compute a priority for different patches so the algorithm proceeds according to a given priority function. In this work, we use the priority function which is defined in [6]. At each step of the algorithm, the border (fill front) of the missing region is identified, and the priority $P(p)$ of each pixel $p$ located on the fill front is calculated. An $n \times n$ patch $\phi_{\hat{p}}$ centered on the highest priority pixel $\hat{p}$ located on the fill front is selected for synthesis. In the patch $\phi_{\hat{p}}$, there are known values and unknown values to be synthesized. A *template* is assumed to be formed with the known pixel values in the input patch $\phi_{\hat{p}}$ (see Fig. 2). The rest is very similar to image prediction case where we search for an optimized approximation of the template by a linear combination of its $K$-NN taken from the known image region, and then keep the same weighting coefficients in the linear combination of the co-located pixels in order to estimate the unknown values. After estimating the unknown sample values in $\phi_{\hat{p}}$, we update the fill front and the algorithm repeats the next steps until all unknown samples in missing region are synthesized.

## 4. EXPERIMENTAL RESULTS

For the experimental validation, we have first placed the proposed prediction method into an image compression scheme. In order to initialize the prediction/compression process, the top 4 rows and left 4 columns of blocks of size $4 \times 4$ are predicted with H.264/AVC intra modes. Once a block has been predicted with the respective prediction method (ATM, NLM, or LLE), the $4 \times 4$ residual block is DCT transformed, quantized, and encoded similar to JPEG. In this coding structure, a uniform quantization matrix with $\Delta = 16$ is weighted by a quality parameter ranging between 10 and 90. Image blocks are processed in the raster scan order, and the reconstructed image is obtained by adding the quantized residue to the prediction. A skip mode has also been included into the encoder to avoid coding the blocks of prediction residue in which all the transformed and quantized coefficients are zero. The corresponding flag is arithmetically encoded. Nine possible forms of *approximation supports* (templates) are considered as shown in Fig. 3. The optimum template type, which is Huffman encoded, is selected by minimizing a rate-

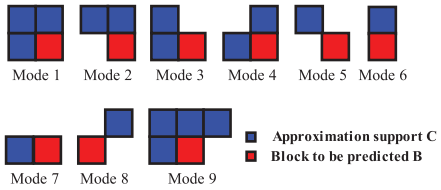


**Fig. 3**. Nine possible forms of approximation support (template). The best type is selected by minimizing the RD cost function.

distortion (RD) cost function of the form $J(\mathbf{D}, \mathbf{R}) = \mathbf{D} + \lambda \mathbf{R}$ where $\mathbf{D}$ is the total distortion (SSE) of the reconstructed block and $\mathbf{R}$ is the estimated encoding cost of the residual signal.

Two sets of tests have been carried out. The first one makes use of approximations with a smaller number of patches considered in the $K$-NN search, where $K$ is varied from 1 to 8. RD optimized $K$ value is signaled to the decoder in addition to the template type. Fig. 5 demonstrates the encoding PSNR/bit-rate performance for Foreman (CIF), Barbara ($512 \times 512$), and Roof ($512 \times 512$) images. The coding cost of the optimum value of $K$ as well as the optimal template type have been included into total bit-rate. One can observe that ATM outperforms the other methods, however LLE has a closer performance than NLM. The second set of experiments relaxes the $K$ value, and takes as $K = 100$. In this case, there is no need to signal the value of $K$ since it is fixed and assumed to be known also at the decoder. Fig. 6 shows the encoding performance results obtained for the same test images. The first observation one can make is that the method based on LLE outperforms the other methods. The significant performance gain can clearly be seen when the image contains more complex and non-periodic texture areas (like in Barbara and Roof). This gain has been achieved because of the bit-rate savings of signaling cost of $K$ (note that this is the same for all methods considered), and of the better optimization solutions of LLE on calculating the weighting coefficients. The second observation is the increase in the performance of NLM in reference to ATM. Apparently, with the increasing number of used image patches, NLM becomes an effective alternative to ATM since the weighting coefficients have been calculated in a wise manner which prevents over-smoothing effects of simply averaging $K = 100$ patches.

We have then placed the proposed texture synthesis method for assessing the performance in a simple loss concealment application. Fig. 4 shows the images (referred to as *barb1*, *barb2*, and *lena*) which have been tested with ATM, NLM, and LLE. Table 1 demonstrates the obtained PSNR results of texture synthesis methods with different $K$ values where $K = \{1, 10, 25, 50, 100\}$. Notice that for the special case when $K = 1$, all of the methods reduce to a simple TM. In terms of PSNR, the effectiveness of LLE method can clearly be observed especially for synthesizing highly textural regions. A linear combination of several texture patches always produce better

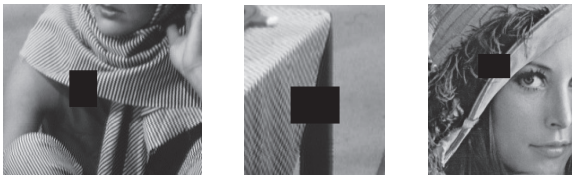

**Fig. 4**. Test images for error concealment: (left) *barb1* (21.25 dB), (middle) *barb2* (18.62 dB), (right) *lena* (21.85 dB).

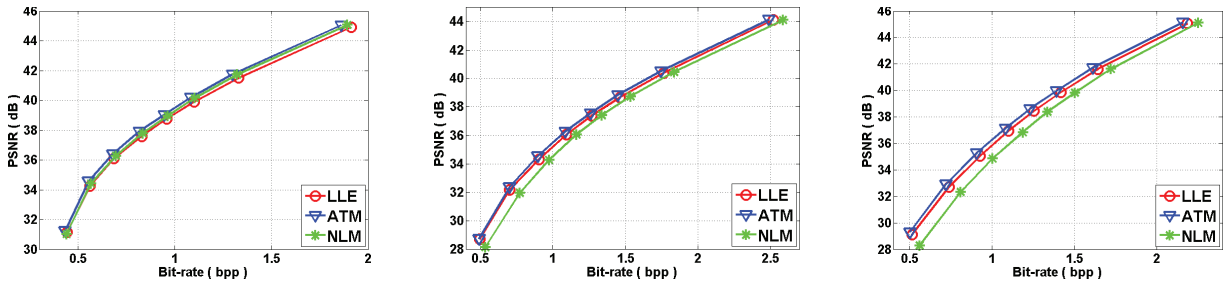


**Fig. 5**. Encoding performance for (left) Foreman, (middle) Barbara, and (right) Roof using LLE in comparison to ATM and NLM (Optimized $K \in [1, 8]$).

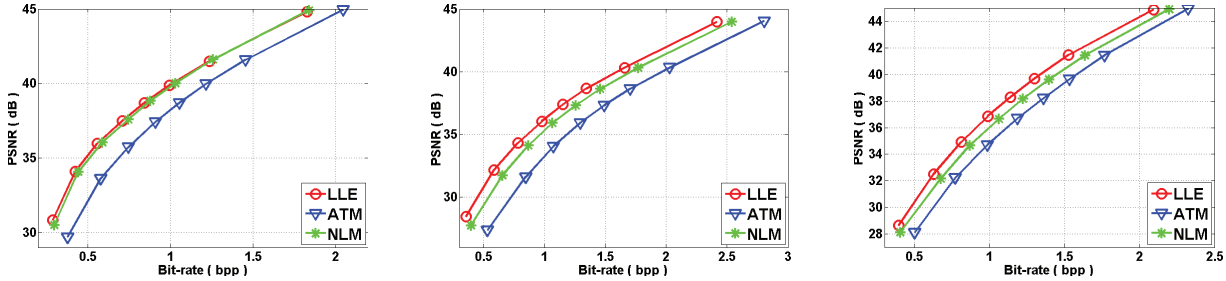


**Fig. 6**. Encoding performance for (left) Foreman, (middle) Barbara, and (right) Roof using LLE in comparison to ATM and NLM (Fixed $K = 100$).

| | TM | ATM | | | | NLM | | | | LLE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Image | $K = 1$ | 10 | 25 | 50 | 100 | 10 | 25 | 50 | 100 | 10 | 25 | 50 | 100 |
| *barb1* | 32.89 | 34.59 | 34.41 | 32.41 | 31.98 | 31.07 | 31.09 | 31.09 | 31.00 | 34.77 | 34.46 | 36.34 | **36.36** |
| *barb2* | 33.70 | 37.24 | 37.41 | 37.68 | 36.73 | 36.35 | 36.41 | 36.70 | 36.71 | **38.05** | 36.21 | 35.19 | 37.78 |
| *lena* | 31.16 | 35.60 | **35.78** | 35.28 | 34.20 | 35.31 | 35.14 | 35.32 | 35.52 | 35.32 | 34.11 | 34.86 | 34.89 |

**Table 1**. PSNR (in dB) results for test images after concealment.

performance when compared to using only one single "best" patch as in TM. Please note that here the tested missing regions are larger than a macroblock size conventionally used for image or video coding, hence one can expect even better results for macroblock size missing regions.

## 5. CONCLUSION

In this paper, we proposed a new texture synthesis method based on LLE. This method can be seen as a generalization of TM, in the sense that it searches to approximate the template signal via a constrained optimization in contrary to heuristic methods such as ATM and NLM. Experimental validation shows that the proposed method offers better performance when compared to TM, ATM, and NLM.

## 6. REFERENCES


[1] M. Bertalmio, A. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proc. IEEE Comp. Soc. Conf. Comp. Vis. Patt. Recog.*, 2001, pp. 355–362.

[2] T. Chan and J. Shen, "Local inpainting models and TV inpainting," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, 2001.

[3] A. 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.

[4] L. Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proc. ACM Comp. Graphics Interactive Tech. (SIGGRAPH 2000)*, 2000, pp. 479–488.

[5] M. Ashikhmin, "Synthesizing natural textures," in *ACM Symp. Interactive 3D Graph.*, 2001, pp. 217–226.

[6] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by examplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[7] 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.

[8] K. Sugimoto, M. Kobayashi, Y. Suzuki, S. Kato, and C. S. Boon, "Inter frame coding with template matching spatio-temporal prediction," in *Proc. IEEE Int. Cong. Image Process.*, 2004, pp. 465–468.

[9] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, Aug. 2003.

[10] T. K. Tan, C. S. Boon, and Y. Suzuki, "Intra prediction by averaged template matching predictors," in *Proc. IEEE Consumer Comm. Network. Conf.*, 2007, pp. 405–409.

[11] A. Wong and J. Orchard, "A nonlocal-means approach to exemplar-based inpainting," in *Proc. IEEE Int. Conf. Image Process.*, 2008, pp. 2600–2603.

[12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE Int. Conf. Computer Vis.*, 2009, pp. 2272–2279.

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

[14] M. Turkan and C. Guillemot, "Image prediction based on neighbor-embedding methods," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1885–1898, Apr. 2012.