IMAGE INPAINTING USING LLE-LDNR AND LINEAR SUBSPACE MAPPINGS

Christine Guillemot, Mehmet Turkan**, Olivier Le Meur***, Mounira Ebdelli**

*: INRIA, Campus Universitaire de Beaulieu, 35042 Rennes, France
**: Technicolor, Cesson-Sevigne, France
***: University of Rennes 1, Campus Universitaire de Beaulieu, 35042 Rennes, France.

ABSTRACT

The paper first describes an examplar-based image inpainting algorithm using a locally linear neighbor embedding technique with low-dimensional neighborhood representation (LLE-LDNR). The inpainting algorithm first searches the K nearest neighbors (K-NN) of the input patch to be filled-in and linearly combine them with LLE-LDNR to synthesize the missing pixels. Linear regression is then introduced for improving the K-NN search. The performance of the LLE-LDNR with the enhanced K-NN search method is assessed for two applications: loss concealment and object removal.

1. INTRODUCTION

Image inpainting refers to the problem of filling-in missing regions in an image [1]. Existing methods can be classified into the following categories. The first category concerns diffusion-based approaches which propagate level lines (called isophotes) using partial differential equations (PDE) [1–4] or variational methods [5]. The second category concerns examplar-based inpainting methods which have been inspired from texture synthesis techniques [6]. These methods exploit image statistical and self-similarity priors. The texture to be synthesized is learned by sampling, copying or by stitching together patches (called examplar) taken from the known part of the image. These methods have evolved over recent years with the introduction of variants related to the patch processing order [7], to fast search of similar patches [8], or to the introduction of spatial coherence constraints [9]. Another category of approaches concerns methods using sparsity priors [10], [11], [12].

Instead of using a simple copy, the unknown pixels of the patch to be filled-in can be approximated by a linear combination of several best matching patches (i.e. of K nearest neighbors, K-NN), this way exploiting self-similarities within the image. The authors in [13] use a similarity kernel to compute the weights of the linear combination. This approach is inspired from the non-local means (NLM) algorithm used for de-noising in [14] and for texture synthesis in [6].

This paper describes an examplar-based inpainting method using linear combinations of neighboring patches in the same vein as in [13]. However, instead of using a similarity kernel, the weights are computed using locally linear embedding with low-dimensional neighborhood representation (called LLE-LDNR in the sequel) [15]. The method is a variant of locally linear embedding (LLE) [16] where the weights are computed on a low-dimensional representation of the neighborhood of the input vector instead of being computed in the high-dimensional input space as in LLE [16]. The algorithm searches for an approximation of the known pixels of the input patch from its K-NN. This principle is known as neighbor embedding (NE).

All NE methods require searching for K-NN. However, the K-NN of the known pixels of the input patch may not lead to good estimates of the unknown pixels. Therefore, we also introduce a method for improving the K-NN search used in the neighbor embedding step. The improved K-NN search makes use of subspace mapping functions learned with linear regression from patches in the known part of the image. The linear regression is also considered for estimating the unknown pixels. The performance of LLE-LDNR and LR-based inpainting solutions is studied in two application contexts: loss concealment and object removal (or image editing), in comparison with state-of-the-art solutions.

2. EXAMPLAR-BASED INPAINTING: BACKGROUND

The proposed algorithm builds upon the solution described in [7] which proceeds as follows. Let I be the image and Ω be the region to be filled in. Let $\phi = I - \Omega$ be the known (or source) region in the image. The algorithm iterates the following steps until all missing pixels have been filled-in:

1. Computation of the priority of each patch located on the fill front as $P(p) = C(p)D(p)$, where the terms $C(p)$ and $D(p)$ are defined as in [7]. The term $C(p)$ is a confidence term based on the ratio of the number of unknown versus known pixels in the input patch. The term $D(p)$ is a data term which aims at favouring patches in which the isophote is perpendicular to the front line in pixel p.

2. Selection of the patch $\Psi_{\hat{p}}$ (centered on a pixel $\hat{p}$ located on the fill front) with the highest priority, i.e.,

$$\hat{p} = \arg\max_{p \in \delta\Omega} P(p).$$

3. Search for the most similar patch Ψ^k_* (in the source image $\phi = I - \Omega$) to the known samples $\Psi^k_{\hat{p}}$ of the input patch $\Psi_{\hat{p}}$ to be filled-in (K-NN search).

4. The unknown pixels of the input patch are set to the values of the pixels at the same relative position in the best matching patch ($\tilde{\Psi}^u_{\hat{p}} = \Psi^u_*$). This method is called *template matching* (TM).

5. Update of the confidence term $C(p)$ used in the patch priority computation.

Instead of using a simple TM, the authors in [13] combine similar patches found in the known part of the image with weights computed with a similarity kernel giving higher weights to patches which are more similar to the known samples of the input patch. Given K-NN patches $\Psi_i, i = 1...\mathrm{K},$, the weights are computed as $\alpha_i = \exp(-\frac{\|\Psi^k_{\hat{p}} - \Psi^k_i\|^2_2}{h})$ where h is a decay coefficient, and then normalized.

3. PROPOSED INPAINTING ALGORITHM

The proposed inpainting procedure differs from [7] summarized above and from [13] in two main aspects: 1/- in the use of a neighbor embedding technique (LLE-LDNR) instead of a simple TM in step 4; and in 2/- the introduction of an enhanced K-NN search using locally learned subspace mapping functions for step 3.

3.1. LLE and LLE-LDNR Neighbor Embedding

The algorithm searches for an approximation of the known part of the input patch $\Psi^k_{\hat{p}}$ from its K-NN $\Psi^k_i, i = 1 \dots K$. The first step is therefore a K-NN search. The K-NN patches $\Psi_i, k = 1...\mathrm{K}$, are formed by pixels denoted Ψ^k_i which are co-located to the known pixels $\Psi^k_{\hat{p}}$ and of pixels Ψ^u_i co-located to the unknown pixels $\Psi^u_{\hat{p}}$. The weights of the linear combination, constrained to sum to "1", can be computed in the input space as in LLE [16], or on the low-dimensional neighborhood representation as in LLE-LDNR [15].

LLE: In [16], the minimizing problem is formulated as

$$\text{argmin } E(W) = \left\| \Psi^k_{\hat{p}} - \sum_i w_i \Psi^k_i \right\|^2_2 \text{ s.t. } \sum_i w_i = 1; \quad (1)$$

Each weight w_i is 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 $\Psi^k_{\hat{p}}$ of the input patch $\Psi_{\hat{p}}$. The i^{th} column of the matrix D is $\Psi^k_i - \Psi^k_{\hat{p}}$, where Ψ^k_i is the i^{th} neighbor of $\Psi^k_{\hat{p}}$. The notation $\mathbf{1_K}$ stands for the column vector of ones of dimension K. 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.

LLE-LDNR: The weights are computed by searching for a rank-d representation of the local neighborhood represented by the matrix D [15]. This is done by computing the singular value decomposition (SVD) of the matrix D as $D = U\Sigma V^T$, where $U = [U_1 U_2]$ and $V = [V_1 V_2]$ are orthogonal matrices of dimension $N_1 \times N_1$ and $K \times K$ respectively, and Σ is a diagonal matrix of dimension $N_1 \times K$ having the singular values of D on its diagonal. The parameter d is the dimension of the low-dimensional space in which are assumed to lie the input data (texture patches). In the experiments reported in Section 4, the parameter d has been taken as $d = \lfloor K/2 \rfloor$, where K is the number of nearest neighbors taken in the embedding.

The matrices U_1 and U_2 contain the first d and last $K - d$ columns of U, and V_1 and V_2 contain the first d and $N_1 - d$ columns of V respectively. The weight vector for the d-dimensional neighborhood of the known samples of the input patch is searched in order to minimize $W^T D_P D_P^T W$ where D_P represents the best rank-d representation of D. The solution is not unique and is taken as the vector in the span of $U_{d+1}, \dots U_K$ that has the smallest l_2 norm leading to the closed form expression [15]

$$w_i = \frac{U_2 U_2^T \mathbf{1_K}}{1_K^T U_2 U_2^T \mathbf{1_K}} \quad (2)$$

LLE captures the data structure in the high dimensional space, whereas LLE-LDNR captures the data structure in the low dimensional space. As a result LLE-LDNR weights are less sensitive to noise. For both methods, once the weights to approximate the known part $\Psi^k_{\hat{p}}$ of the input patch have been found, they are applied to estimate the unknown part as

$$\Psi^u_{\hat{p}} = \sum_{i=1}^{K} w_i \Psi^u_i \quad (3)$$

3.2. K-NN search using learning subspace mappings

The K-NN to the known pixels of the input patch may not be the best patches for approximating the unknown pixels, especially in the case where there are discontinuities within the patch. In order to have K-NN patches which are relevant for the unknown part, the idea is to use mapping functions between subspaces corresponding to *known* Ψ^k_i and *unknown* Ψ^u_i parts of complete patches.

Let X be a matrix whose columns are vectors of dimension N (size of the complete patch) formed by all candidate patches $\Psi_i, i = 1 \dots L$ (training patches), taken from a search window in the known part of the image. The parameter L denotes the number of patches in the search window. Let X_k and

X_u be matrices whose columns are vectors of dimensions N_1 and N_2 formed by the subsets Ψ_i^k and Ψ_i^u of pixels located in Ψ_i at the same respective positions as the known and unknown pixels of the patch to be filled-in. Two mapping functions $F_1 : X_k \xrightarrow{F_1} X_u$ and $F_2 : X_k \xrightarrow{F_2} X$ are learned using multivariate linear regression as

$$F_1 = X_u X_k^T (X_k X_k^T)^{-1} \quad (4)$$

$$F_2 = X \;\; X_k^T (X_k X_k^T)^{-1} \quad (5)$$

The known part $\Psi_{\hat{p}}^k$ of the input patch is projected (or mapped) via F_1 into $\mathbf{R}^{N_2}$ as

$$\Psi_{\hat{p}}^k \in \mathbf{R}^{N_1} \xrightarrow{F_1} \tilde{\Psi}_{\hat{p}}^u \in \mathbf{R}^{N_2} \quad (6)$$

This projection actually computes an estimate $\tilde{\Psi}_{\hat{p}}^u = F_1 \times \Psi_{\hat{p}}^k$ (via linear regression) of the unknown pixels of the input patch. This estimate can be used directly as the inpainted pixels, leading to a method referred to as Linear Regression (LR)-based inpainting in the sequel. It can also be used to help the K-NN search needed in the neighbor embedding.

Instead of considering only the K-NN to the known pixels $\Psi_{\hat{p}}^k$ (which may not be good candidates for approximating the unknown part of the input patch), we also search for the K-NN to the LR estimate $\tilde{\Psi}_u$. Among the resulting $2K$ patches, we retain, for applying the neighbor embedding, the K patches which are the closest to $\tilde{\Psi}_{\hat{p}}$ which results from the projection $\Psi_{\hat{p}}^k \xrightarrow{F_2} \tilde{\Psi}_{\hat{p}}$.

4. EXPERIMENTAL RESULTS

The methods have been assessed in the context of two applications: loss concealment and object removal. In the object removal application, the ground truth is not known. The performances can, in this case, only be assessed visually, the inpainted images must look as natural as possible. In the experiments, each processed patch is of size 11×11. For the methods combining neighboring patches (NLM, and all the other NE methods), the maximum number of patches considered has been set to $K_{max} = 10$. However, the actual number of neighbors used varies from one patch to the other. One keeps only the most similar patches for which the distance to the input patch $\Psi_{\hat{p}}$ is below $d_1 + \delta \times d_1$, where d_1 is the distance of the most similar patch. In the experiments, the δ parameter has been set to 1.5.

4.1. Loss concealment

Fig. 1 shows the test images for the loss concealment scenario. The missing blocks are of size 32×32. The size of the K-NN search window (centered on the patch to be filled) has been set to 100×100 for the House image and to 60×60 for the smaller images Barbara and Foreman. Table 4.1 gives the PNSR values obtained when inpainting the test images shown in Fig.1, with TM, NLM, LLE, LLE with enhanced KNN search (noted LLE-LR), LLE-LDNR, and LLE-LDNR with enhanced K-NN search (noted LLE-LDNR-LR). It also gives the results obtained when using linear regression (LR) to directly estimate the unknown pixels. All the results have been produced in an automatic way (with no supervision). The right column of the table gives the results obtained with a diffusion-based method [3]. The code used for simulating the method in [3] is the openCV implementation. This table shows superior performance of the LLE-LDNR solution compared to the other NE solutions. Its superior performance compared to LLE results from a more robust embedding (weight computation) capturing dominant features in the low-dimensional space. It also shows the gain brought by the enhanced K-NN search. Fig. 2 shows the visual quality obtained with some of the methods.

4.2. Object removal

In the context of object removal (image editing application), we have observed that a better visual quality was obtained by limiting the size of the search window to a small value (40×40). Fig. 2 shows inpainting results obtained for the Terrasse image with the different solutions. The most natural inpainted images have been obtained with the LLE-LDNR neighbor embedding method. Since the holes to be filled-in are quite large, the inpainting task is not easy and diffusion-based approaches introduce blur into the image. When we compare the proposed methods with other exemplar-based methods including [7] and [8], more natural looking images with better preserved structures have been obtained. The method based on low rank tensor completion proposed in [17] has also been considered for comparison. However, this method works well only when the missing area is not too large or when the rank of the image is quite low, which is in general not the case for loss concealment or object removal scenarios.

5. CONCLUSION

This paper describes an image inpainting algorithm using LLE-LDNR neighbor embedding. In addition, the paper considers linear regression for improving the K-NN search as well as for estimating unknown pixels. Experimental results in two applications (loss concealment and object removal) show superior performances of the LLE-LDNR solution over other neighbor embedding solutions. Although, only comparisons with NLM and LLE are given here, tests have shown better performance than with Non-Negative Matrix Factorization as well. The results also show further gains when using the proposed enhanced K-NN search using linear subspace mappings in the context of inpainting.

Acknowledgement: This work has been partly funded by the ANR-ARSSO project.

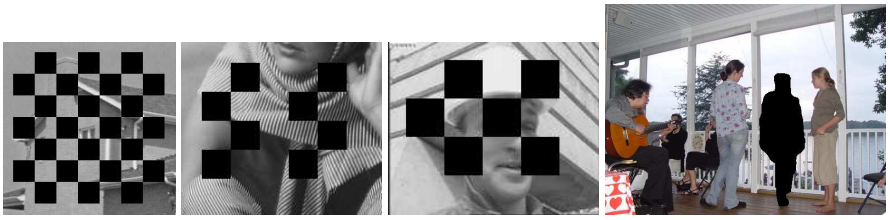

Fig. 1. Test images with holes (black rectangles). The three first images called House (8.71 dB with the holes), Barbara (13.37 dB with the holes), and Foreman (9.92 dB with the holes) are used for the loss concealment scenario and the last one called Terrasse is used for the object removal scenario.

Methods	TM [7]	NLM	LLE	LLE-LR	LLE-LDNR	LLE-LDNR-LR	LR	diff. [3]
House	24.07	24.51	25.05	25.05	26.18	**26.44**	26.19	25.30
Barbara	22.68	23.06	23.86	24.54	24.42	24.93	**25.01**	23.87
Foreman	24.35	24.68	25.01	25.47	27.08	**27.80**	25.30	25.22

Table 1. PSNR (in dB) in the context of loss concealment (patch size w=11×11, and $\delta = 1.5$).

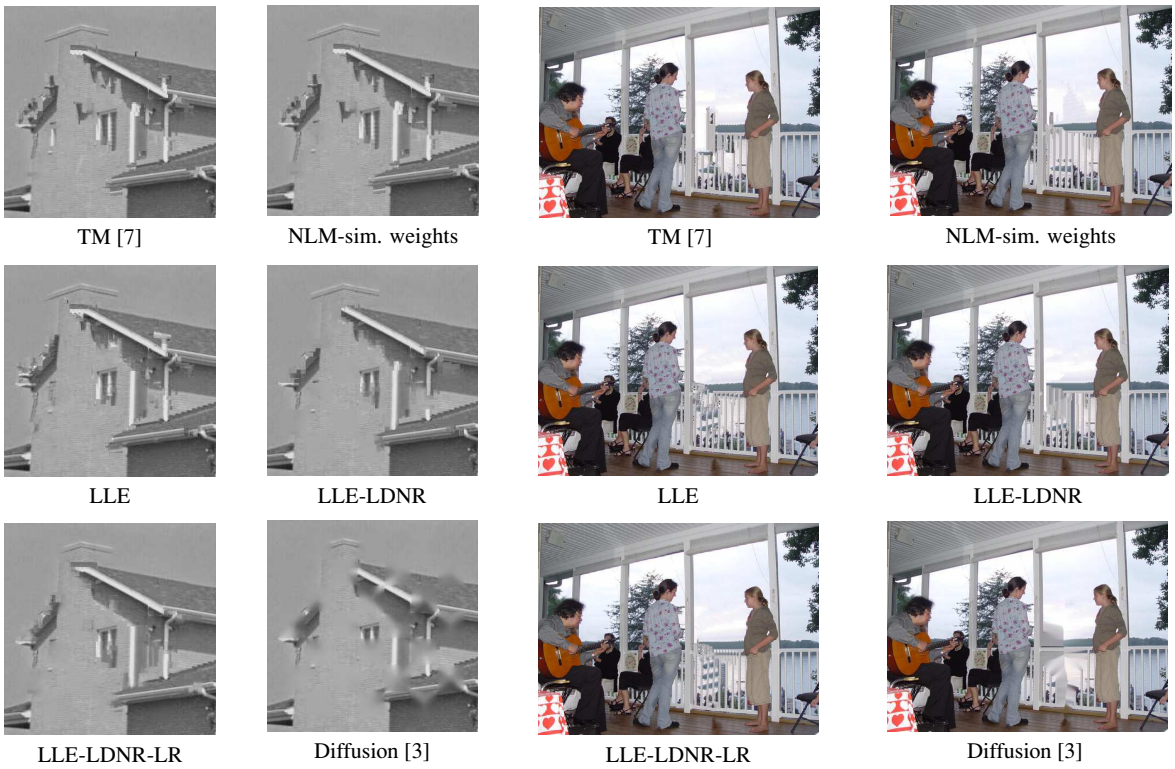

Fig. 2. Results for the House image (loss concealment) and for the Terrasse image (object removal).

6. REFERENCES

[1] M. Bertalmio, G. Sapiro, C. Ballester, and V. Caselles, "Image inpainting," in *Proc. ACM SIGGRAPH*, Jul. 2000, pp. 417–424.

[2] M. Bertalmio, A. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proc. CVPR*, Dec. 2001, pp. 355–362.

[3] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graphics Tools*, vol. 9, no. 1, pp. 25–36, 2004.

[4] D. Tschumperle, "Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's," *Int. J. Comp. Vision*, vol. 68, no. 1, pp. 65–82, 2006.

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

[6] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conf. on Computer Vision*, Sep. 1999.

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

[8] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graphics*, vol. 28, no. 3, 2009.

[9] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[10] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Trans. on Image Processing*, vol. 19, no. 5, pp. 1153–1165, May 2010.

[11] C. Studer, P. Kuppinger, G. Pope, and H. Bolckei, "Recovery of sparsely corrupted signals," *IEEE trans. on Information Theory*, vol. 58, no. 5, pp. 3115 – 3130, May 2012.

[12] O. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising - part i: Theory," *IEEE Trans. on Image Processing*, vol. 15, no. 3, pp. 539–554, Mar. 2006.

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

[14] 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.

[15] 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.

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

[17] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012.