GUIDED INPAINTING WITH CLUSTER-BASED AUXILIARY INFORMATION

Thomas Maugey[1], *Pascal Frossard*[2], *Christine Guillemot*[1]

[1]INRIA Rennes-Bretagne-Atlantique [2]Ecole Polytechnique Fédérale de Lausanne (EPFL)

ABSTRACT

In this paper, we propose a new guided inpainting algorithm based on the exemplar-based approach in order to effectively fill in holes in image synthesis applications. Guided inpainting techniques can be very useful in settings where one has access to the ground truth information like most multiview coding applications. We propose a new auxiliary information based on patch clustering, which is used to refine the candidate exemplar set in the inpainting. For that purpose, a new recursive clustering method based on locally linear embedding (LLE) is introduced. We then design the guided inpainting solution based on LLE with clustered patches, which contrains the reconstruction to operate in one patch cluster only. The index of the appropriate cluster considered as auxiliary information. Experimental results show that our clustering algorithm provides clusters that are well suited to the inpainting problem. They also show that the auxiliary information enables to significantly improve the quality of the inpainted image for a small coding cost. This work is the first study to show that effective inpainting can be performed when the auxiliary information is properly adapted to the characteristics of both the hole and the known texture.

Index Terms— Guided inpainting, auxiliary information, clustering

1. INTRODUCTION

Inpainting of images has been intensively studied in the past few years, especially for applications such as image restoration and editing [1]. Another application where inpainting techniques are useful is view synthesis, where holes are to be filled corresponding to areas that are no longer occluded. In the particular cases where one has access to ground truth images (like for example in multiview video coding where view synthesis is used for predicting the captured views from a reference one), auxiliary information can be generated to help inpainting, which leads to the concept of *guided inpainting*.

The few guided inpainting techniques that have been developed in the literature are exemplar-based inpainting methods and differ in the auxiliary information. In [2], the hole filling process is helped with image structure information such as edges. In [3], the auxiliary information is made of the DC coefficients of the patches to be filled and a DC coefficient comparison term is taken into account in the exemplar-based patch reconstruction. In [4], the auxiliary information takes the form of inpainting modes, as in standards video codecs [5], where the encoder can choose between “intra”, “vector” and “skip” modes. All these techniques achieve promising performances and validate the idea that inpainting can benefit from side information. Unfortunately, the auxiliary information built in these methods does however not depend on the known region, but is only based on the ground truth information corresponding to the hole, despite the fact that the exemplar-based inpainting methods use the known part of the image for filling.

In this paper, we address this limitation and we propose a new auxiliary information that is used to refine the set of candidate patches for the hole filling step of the inpainting. We first assume that patches of images lie in a union of subspaces, *i.e.*, the images have different regions with different color textures. We thus cluster the image patches using a new recursive spectral clustering algorithm that extends the sparse subspace clustering of [6] and replaces the sparse approximation by locally linear embedding, better suited for the inpainting context. Dictionaries are then built from these clusters and used for the hole filling process. However, the inpainting is not always able to guess in which cluster the patches of the hole belong to (especially around discontinuities). The auxiliary information that is built from the ground truth image may help to find the right cluster. We thus propose a new guided inpainting algorithm that forces the patch reconstruction to be done in one cluster only, if no auxiliary information is available, or in the cluster pointed by the auxiliary information, if it is available. Experimental results show that the clusters obtained with our method are well adapted to the inpainting problem. Moreover, they confirm that the auxiliary information improves significantly the quality of inpainted images for small rate overhead.

The remainder of the paper is organized as follows. In Sec. 2, we detail the guided inpainting framework and the main lines of our solution. In Sec. 3, we introduce the new patch clustering method. Then in Sec. 4, we detail our auxiliary information based inpainting algorithm.

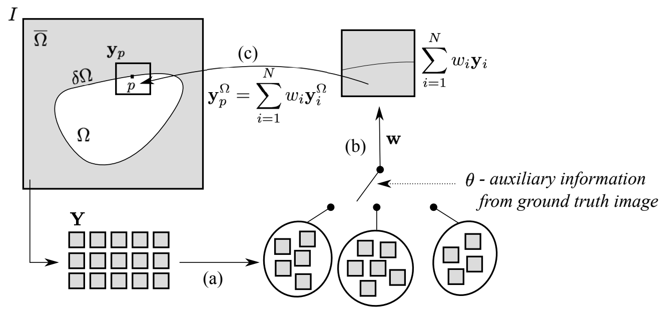
2. GUIDED INPAINTING FRAMEWORK

Let I be an image and $\Omega \subset I$ be the region to be inpainted. Let $\overline{\Omega} = I \setminus \Omega$ be the known region and $\delta\Omega$ be the border between the known and unknown regions. The mostly adopted exemplar-based inpainting algorithms ([7], further extended in [8]) proceed patch per patch and alternate between two main steps, namely i) the patch priority calculation and ii) the patch filling using patches of the known region. The purpose of step i) is to select the patch $\mathbf{y}_p$, centered on a pixel p, to be filled in by step ii)[1]. The main challenge is to start by inpainting patches that are easy to inpaint and that do not lead to high propagation error. The idea is to firstly select the patches having strong structures (*e.g.*, edges). For that purpose, the selection algorithm calculates, for each candidate patch $\mathbf{y}_{p'}$ located on the border $\delta\Omega$, a priority term defined in [8] as

$$P(p') = C(p')D(p')E(p'). \tag{1}$$

This priority is the product of three functions, respectively the confidence, the data and the edge terms. The confidence term $C(p')$ is defined as the portion of known pixels in the patch $\mathbf{y}_{p'}$. The data term $D(p')$ is the inner product between the isophote direction and

[1]In the following, a patch is represented as a vector of its color values.



I
$\overline{\Omega}$
$\mathbf{y}_p$
$\delta\Omega$
p
Ω
(c)
$\mathbf{y}_p^{\Omega} = \sum_{i=1}^{N} w_i \mathbf{y}_i^{\Omega}$
$\sum_{i=1}^{N} w_i \mathbf{y}_i$
(b)
$\mathbf{w}$
θ - auxiliary information from ground truth image
$\mathbf{Y}$
(a)

Fig. 1. Proposed guided inpainting framework with (a) the dictionary $\mathbf{Y}$ clustering, (b) the vector $\mathbf{w}$ construction with the help of the AI θ, (c) the patch filling using Eq. (3).

the normal of the border. It gives priority to patches that contain edges perpendicular to the border. Finally, the edge term $E(p')$ calculates the ratio of pixels of $\mathbf{y}_{p'}$ belonging to an edge. In step ii), the algorithm reconstructs the selected patch $\mathbf{y}_p$, or more exactly its unknown region $\mathbf{y}_p \cap \Omega$ (denoted by $\mathbf{y}_p^{\Omega}$)[2]. This reconstruction step uses the patches of the known region to build an estimation whose known part is close to $\mathbf{y}_p^{\overline{\Omega}}$. Several methods have been considered for this estimation: copy [7] or linear combination of nearest neighbors with least square approximation [8]. In the latter case, a set of fully known patches called $\mathcal{Y}$ is available, made of N patches $\mathbf{y}_i$. They form a matrix $\mathbf{Y}$. The inpainting algorithm has to determine the linear combination coefficients $\mathbf{w}$ that give the best representation of $\mathbf{y}_p^{\overline{\Omega}}$, *i.e.*, with the smallest approximation error. It solves the following problem:

$$\mathbf{w} = \operatorname{argmin}_{\boldsymbol{\alpha}} ||\mathbf{y}_p^{\overline{\Omega}} - \sum_{i=1}^{N} \alpha_i \mathbf{y}_i^{\overline{\Omega}}||_2^2 \quad \text{s.t.} \quad \sum_{i=1}^{N} \alpha_i = 1. \tag{2}$$

Then, $\mathbf{y}_p^{\Omega}$ is filled as follows

$$\mathbf{y}_p^{\Omega} = \sum_{i=1}^{N} w_i \mathbf{y}_i^{\Omega}. \tag{3}$$

These approaches may fail for example if texture discontinuities occur on $\delta\Omega$ in the ground truth image. This reconstruction step ii) would thus greatly benefit from a guidance in that situation. In the scenario where one has access to the ground truth Ω (*e.g.*, coding applications), this guidance can be done with some *auxiliary information* (AI) θ. Such AI-based algorithms are called *guided inpainting*. In exemplar-based inpainting context, that AI could refine the candidate patch set, *i.e.,* it could identify subsets of well-chosen exemplars that should be used in the reconstruction. These subsets can be obtained by clustering similar patches together. The auxiliary information then indicates to the inpainting process which cluster has to be used for the patches whose construction is problematic. The proposed guided inpainting framework is summarized in Fig. 1.

3. PATCH CLUSTERING

We now introduce the clustering algorithm that is used in our proposed guided inpainting method (step (a) in Fig. 1). The proposed sparse subspace clustering algorithm has two main steps. First, a similarity graph is constructed. This graph $\mathcal{G}$ connects the patches $\mathbf{y} \in \mathcal{Y}$ if they are similar. The edge set is denoted by $\mathcal{E}$. The weights $\mathcal{W}$ give a measure of this similarity. Second, the graph is cut in different sub-graphs that define the clusters (*i.e.,* the sub-dictionaries in the inpainting). The clustering technique extends the method proposed in [6] by considering a locally linear embedding rather than using matching pursuit algorithms [9] for the construction of $\mathcal{W}$. Another difference is that we cluster the graph $\mathcal{G}$ using a recursive algorithm instead of the spectral clustering used in [6].

3.1. Similarity graph construction

Two patches are said similar, if one appears in the sparse representation of the other one. In other words, for each patch $\mathbf{y}_i$, one estimates its best approximation using other patches $\mathbf{y}_j$ ($j \neq i$). The weights are the energy of the linear combination coefficients. The estimation is based on locally linear embedding (LLE) introduced in [10]. It has been shown that the LLE algorithm gives good results in the specific context of inpainting [1]. More precisely, the algorithm solves the following criterion that aims at finding the optimal coefficients of the linear combination $\mathbf{c}$, for a given $\mathbf{y}_i$:

$$\begin{aligned} \mathbf{c} \quad &= \operatorname{argmin}_{\boldsymbol{\alpha}} ||\mathbf{y}_i - \mathbf{Y}\boldsymbol{\alpha}|| \\ &\text{s.t.} \quad \textstyle\sum_{j=1}^{k} \alpha_j = 1, \\ &\text{and} \quad \alpha_i = 0. \end{aligned} \tag{4}$$

The error can be rewritten as $E(\mathbf{c}) = \mathbf{c}^T \mathbf{B}^T \mathbf{B} \mathbf{c}$, where $\mathbf{B}$ is a matrix whose j-th column is $\mathbf{y}_i - \mathbf{y}_j$. As it is done in [10], the vector $\mathbf{c}$ is thus the solution of the linear system $\mathbf{B}^T \mathbf{B} \mathbf{c} = \mathbf{1}_k$. The weights $\mathbf{c}$ are then rescaled so that their sum is 1. We define $\mathbf{C}$ as the matrix whose i-th column corresponds to the representation of $\mathbf{y}_i$ after solving the problem (4). We define the adjacency matrix of the graph $\mathcal{G}$ as $\mathbf{A} = \mathbf{C} + \mathbf{C}^T$.

3.2. Recursive spectral clustering

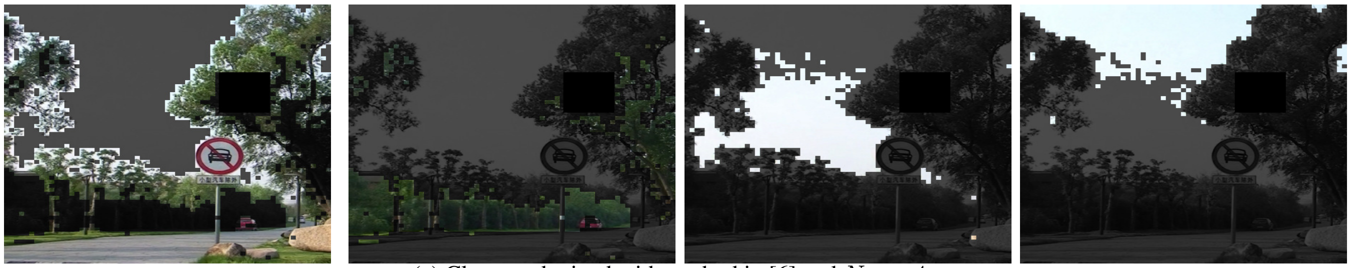
In this work, we propose a *recursive spectral clustering* (RSC), that refines the spectral clustering (SC). The SC has been proposed in [11] and works as follows. First the laplacian $\mathbf{L}$ of the graph is computed as

$$\mathbf{L} = \mathbf{D} - \mathbf{A},$$

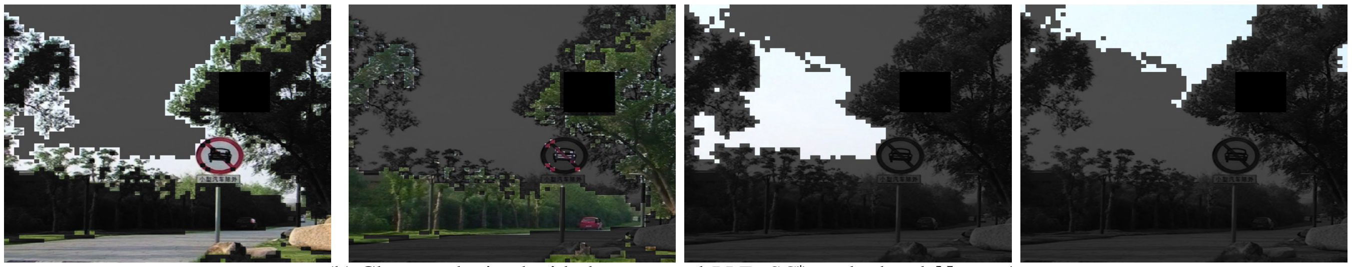
where $\mathbf{D}$ is the degree matrix of the graph, *i.e.,* the diagonal matrix whose i-th diagonal element is equal to the sum of the weights of all the edges inherent to vertex i. Then, we compute the N_K eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_j, \dots, \mathbf{v}_{N_K}$ of the matrix $\mathbf{L}$. Let $\mathbf{V}$ be the matrix whose columns are the vectors $\mathbf{v}_j$. The clustering is then done by applying a k-means [12] algorithm on the rows of $\mathbf{V}$, with N_K clusters.

The spectral clustering separates the vertices with respect to their connectivity relationship. We show an example of clustering obtained with this spectral clustering in Fig. 2 (for $\mathbf{A}$ estimated with (a) matching pursuit algorithm and (b) LLE). From the clustered representation of the adjacency matrix $\mathbf{A}$ shown in Fig. 3(a), we see that the algorithm groups satisfactorily the patches that are connected together. However, if we look at the clusters themselves (Fig. 2(a) and (b)), we notice that some homogeneous clusters have been divided into two subspaces, while the two first clusters are still heterogeneous. This can be explained by the fact that the arbitrarily chosen N_K given to the spectral clustering algorithm does not fit with the actual number of clusters present in the image. We propose to face this problem by relying on the averaged approximation

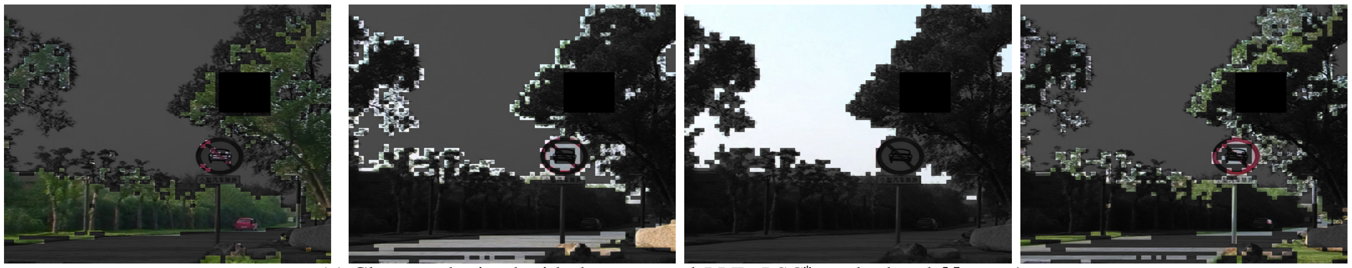
[2]On the contrary the known part of $\mathbf{y}_p$ is written as $\mathbf{y}_p^{\overline{\Omega}}$. The notation is generalized to any vectors representing a patch.



(a) Clusters obtained with method in [6] and $N_K = 4$

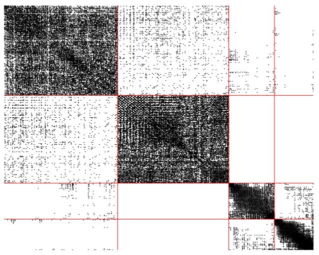


(b) Clusters obtained with the proposed *LLE+SC** method and $N_K = 4$

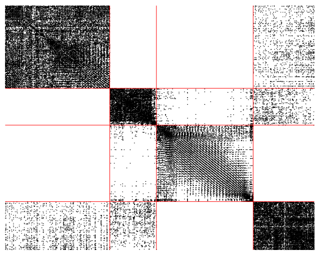


(c) Clusters obtained with the proposed *LLE+RSC** method and $N_K = 4$

Fig. 2. Clustering results for *road* image obtained with SC, for $N_K = 4$ for the three following methods: (a) [6] where $\mathbf{A}$ is built using matching pursuit and is clustered using SP, proposed (b) *LLE+SC** and (c) *LLE+RSC** where $\mathbf{A}$ is built using LLE and is clustered using respectively SC and RSC.



(a)



(b)

Fig. 3. Clustered $\mathbf{A}$ matrix for *road* image and $N_K = 4$ using (a) *LLE+SC**, and (b) *LLE+RSC** methods.

error within each cluster, in order to force the cluster homogeneity although N_K is inferior to its true value.

The idea of proposed RSC is to start performing a SC with $N_K = 2$, and then, to subdivide the cluster that has the highest reconstruction mean square error (MSE). For the same test image as SC, we have run the RSC algorithm, and we see in Fig. 2(c), that the clustering subdivides heterogeneous regions. The clustering remains satisfactory as shown in Fig. 3(b). For sake of space saving, we do not show other image examples, but a similar conclusion has been drawn with other test images.

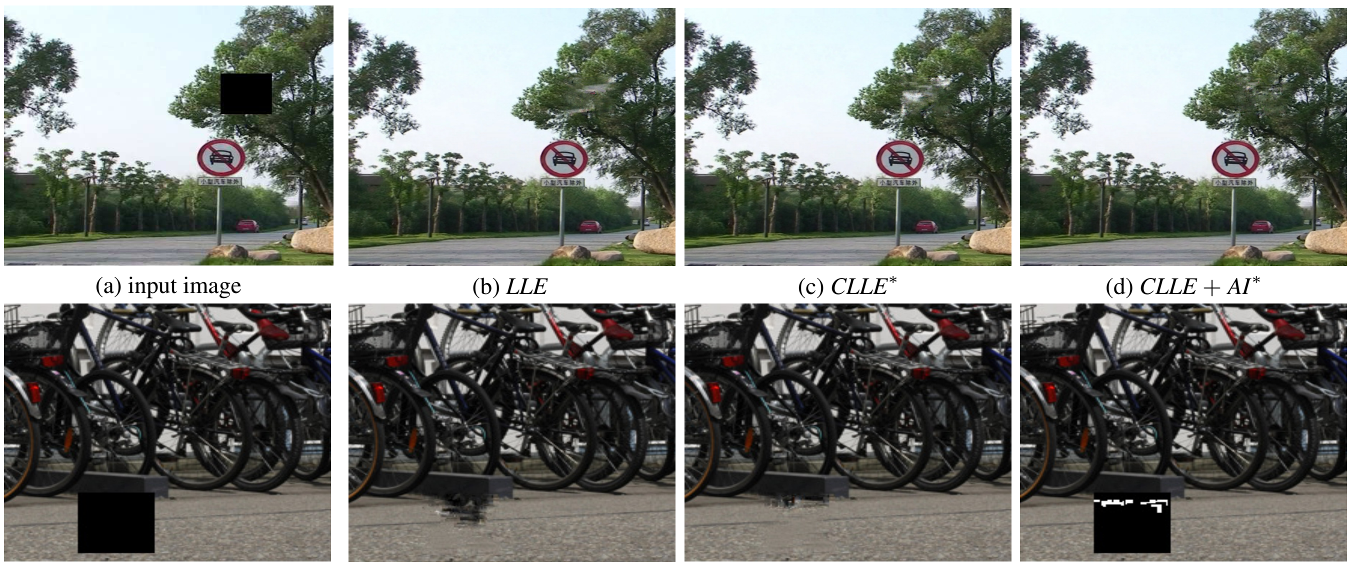
4. CLUSTER-BASED GUIDED INPAINTING

We first define two entities that have two different roles: the *Auxiliary Information Generator (AIG)* that has access to the ground truth image and construct the AI, and the *Inpainting Executor (IE)* that inpaints the hole based on the knowledge of the AI. Typically in coding applications, the IE is at the decoder side while the AIG is at the encoder.

Let us first explain how the AIG works. We first form a set $\mathcal{Y}$ containing the whole set of known patches. This patch set is clustered into N_K sub-dictionaries $\mathcal{Y}_i$ with the algorithm introduced in the previous section. The indices of the patches in the sub-dictionary $\mathcal{Y}_i$ are grouped in a set called S_i. We denote by $\mathbf{Y}_i$, $\mathbf{Y}_i^{\Omega}$ and $\mathbf{Y}_i^{\overline{\Omega}}$ the matrices whose columns are respectively the patches in $\mathcal{Y}_i$, the unknown and known regions of these patches[3]. For a given patch $\mathbf{y}_p$ to be filled in (*i.e.,* $\mathbf{y}_p^{\Omega}$ is known while $\mathbf{y}_p^{\overline{\Omega}}$ is unknown), the algorithm firstly selects the k nearest neighbors by choosing the patches of $\mathcal{Y}_i^{\overline{\Omega}}(p)$ that have the smallest distance with $\mathbf{y}_p^{\overline{\Omega}}$. Let $d_{\min}$ be the distance of the closest neighbour of $\mathbf{y}_p^{\overline{\Omega}}$[4]. The algorithm then selects the k patches whose distance with $\mathbf{y}_p^{\overline{\Omega}}$ is smaller than $\beta d_{\min}$ (with $\beta > 1$). The indices of these patches are grouped in a set called

[3]For sake of clarity, we have simplified the notations, because the dictionaries $\mathbf{Y}^{\Omega}$ and $\mathbf{Y}^{\overline{\Omega}}$ depend on the mask shape, hence on p.

[4]$d_{\min}$ is computed with the mean square distance.



(a) input image (b) *LLE* (c) *CLLE** (d) *CLLE* $+$ *AI**

(e) input image (f) *CLLE** (g) *CLLE* $+$ *AI** (h) AI position for *CLLE* $+$ *AI**

Fig. 4. Visual comparison between inpainted images using *LLE*, *CLLE** and *CLLE* $+$ *AI** methods, for *road* (455×455) and *bikes* (217×217) images.

$\mathcal{K}_i(p) \subset S_i$. Within each set $\mathcal{K}_i(p)$, the problem thus becomes:

$$\begin{aligned} \mathbf{w}_i^{\overline{\Omega}} &= \operatorname{argmin}_{\boldsymbol{\alpha}} ||\mathbf{y}_p^{\overline{\Omega}} - \mathbf{Y}_i^{\overline{\Omega}}\boldsymbol{\alpha}|| \\ \text{s.t.} &\quad \textstyle\sum_{j=0}^{k} \alpha_j = 1, \\ \text{and} &\quad \forall j \notin \mathcal{K}_i, \quad \alpha_j = 0. \end{aligned} \tag{5}$$

As it is done for Eq. (4) and in [10], the error can be rewritten as $E(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T\mathbf{B}^T\mathbf{B}\boldsymbol{\alpha}$, where $\mathbf{B}$ is a matrix whose j^{th} column is $\mathbf{y}_p^{\overline{\Omega}} - \mathbf{y}_{\mathcal{K}_i(j)}^{\overline{\Omega}}$. The vector $\mathbf{w}_i^{\overline{\Omega}}$ is thus the solution of the linear system $\mathbf{B}^T\mathbf{B}\boldsymbol{\alpha} = \mathbf{1}_k$. The weights $\mathbf{w}_i^{\overline{\Omega}}$ are then rescaled so that their sum is 1. The optimal cluster index $i_*^{\overline{\Omega}}$, *i.e.,* the one yielding the best approximation, is derived from

$$i_*^{\overline{\Omega}} = \operatorname{argmin}_i ||\mathbf{y}_p^{\overline{\Omega}} - \mathbf{Y}_i^{\overline{\Omega}}\mathbf{w}_i^{\overline{\Omega}}||. \tag{6}$$

The cluster $i_*^{\overline{\Omega}}$ is in fact the one that will be chosen by the IE. We then compute the true cluster index i_*, *i.e.,* the one taking into account the whole patch information $\mathbf{y}_p$. Eq. (5) and (6) respectively become $\mathbf{w}_i = \operatorname{argmin} ||\mathbf{y}_p - \mathbf{Y}_i\boldsymbol{\alpha}||$ and $i_* = \operatorname{argmin}_i ||\mathbf{y}_p - \mathbf{Y}_i\mathbf{w}_i||$. In order to guide the IE and to make it use the good set of patches for the reconstruction, the AI θ is:

$$\theta = \begin{cases} 0 & \text{if } i_*^{\overline{\Omega}} = i_* \\ i_* & \text{if } i_*^{\overline{\Omega}} \neq i_*. \end{cases} \tag{7}$$

In Eq. (7), the AI is non-zero only if the inpainting is not able to find the optimal cluster i_* from the known part of the patch only.

The IE first builds the same N_K cluster as the AIG. At the reconstruction step, for a patch whose attached auxiliary information is 0, the IE finds the optimal cluster index using only the known part of the image, *i.e.,* using Eq. (5) and (6). On the contrary, if the AI points the right cluster θ, the coefficient vector $\mathbf{w}$ is directly determined with Eq. (5) in the appropriate cluster $i = \theta$. The patch is finally computed using (3).

We illustrate the benefits of the proposed approach with two test images called *road* (RGB, 455×455) and *bikes* (RGB, 217×217). For the hole Ω, we run the three following inpainting techniques (one baseline and two proposed algorithms denoted by *): (1) *LLE* is the inpainting proposed in [8] (*i.e.,* without patch clustering), (2) *CLLE** is our *LLE* inpainting method with clustered patches ($N_K = 4$) with all $\theta = 0$, and (3) *CLLE* $+$ *AI** is the *CLLE**, where the inpainting is guided with the cluster information as AI. Results are presented in Fig. 4. We first see from Fig. 4(b) and (c) that the clustering of the patches already helps to improve the reconstruction quality. More precisely, it reduces the smoothness of the inpainted region (visually unpleasant) as it is visible in Fig. 4(b) for the *LLE* method. Then, we observe that the use of AI leads to an even better inpainting quality. More precisely, for *bikes* image, we see that the AI forbids the inpainting to propagate the dark texture towards the ground. This is even more visible in Fig. 4(h), where the blocks for which the AI is sent are displayed in white. We observe that they correspond to the position where *CLLE** starts to fail (see Fig. 4(g)). For coding application, the AI is generated at the encoder and transmitted to the decoder. The use of AI is interesting if the rate overhead remains reasonable. We have coded the vector of indices θ in the example of Fig. 4, and we have obtained 0.018 bpp, which is negligible compared to typical image compression rates.

5. CONCLUSION

In this paper, we have presented a new guided inpainting method. This method relies on an analysis of the known region of the image. The available texture is clustered, and the inpainting is constrained so that it works in only one of these clusters. The AI considered for a subset of the inpainted patches is the cluster index, and tells to the inpainting to what texture class belongs the current patch to be filled. The visual results show a significant improvement. For coding applications, the rate overhead is small. This work opens perspectives for guided inpainting, and highlight the importance taking into account the known image in the AI design.

6. REFERENCES

[1] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *Signal Processing Magazine*, vol. 31, no. 1, pp. 1053–5888, Dec. 2013.

[2] C. Wang, X. Sun, F. Wu, and H. Xiong, "Image compression with structure-aware inpainting," in *IEEE Int. Symp. on Circuits and Systems*, Island of Kos, May 2006.

[3] T. Maugey, P. Frossard, and G. Cheung, "Consistent view synthesis in interactive multiview imaging," in *Proc. IEEE Int. Conf. on Image Processing*, Orlando, Florida, US, Oct. 2012.

[4] I. Daribo, T. Maugey, G. Cheung, and P. Frossard, "R-D optimized auxiliary information for inpainting-based view synthesis," in *IEEE 3DTV-Conference*, Zurich, Switzerland, 2012.

[5] K. Müller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, H. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Trans. on Image Proc.*, vol. 22, no. 9, pp. 3366 – 3378, May 2013.

[6] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. on Pattern Anal. and Match. Int.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

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

[8] C. Guillemot, M. Turkan, O. Le Meur, and M. Ebdellia, "Object removal and loss concealment using neighbor embedding methods," *Signal Processing: Image Communication*, vol. 28, no. 10, pp. 1405–1419, Nov. 2013.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[10] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[11] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Neural Information Processing Systems*, pp. 849–856, 2001.

[12] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley-Interscience, 2006.