# AN AUTOMATIC STRUCTURE-AWARE IMAGE EXTRAPOLATION APPLIED TO ERROR CONCEALMENT

*Haricharan Lakshman*[1]*, Patrick Ndjiki-Nya*[1]*, Martin Köppel*[1]*, Dimitar Doshkov*[1] *and Thomas Wiegand*[1,2]

[1]Image Processing Department
Fraunhofer Institute for Telecommunications - Heinrich Hertz Institute
Einsteinufer 37, 10587 Berlin, Germany

[2]Image Communication Chair
Department of Telecommunication Systems
Technical University of Berlin
Einsteinufer 17, 10587 Berlin, Germany

## ABSTRACT

A novel framework for spatially estimating unknown image data is presented. Common applications include inpainting, concealment of transmission errors, prediction in video coding, etc. Firstly, a segmentation of the spatial neighborhood of the area to be estimated is performed and a plausible set of segments that cross the unknown area is identified. Then, a reconstruction algorithm is developed by combining sparse modeling and patch-based synthesis. The improved extrapolation capabilities of the presented approach is shown for variety of image characteristics and the robustness of the algorithm is illustrated for large unknown blocks, which are becoming especially important for future video coding standards in order to efficiently code high resolution content.

***Index Terms***— Error concealment, Texture synthesis, Extrapolation, Inpainting

## 1. INTRODUCTION

The process of extending a discrete signal from known areas into unknown areas is called signal extrapolation. Transmission of images or videos in error prone environments may lead to block losses. In order to estimate the missing image data, spatial extrapolation can be applied in image transmission, intraframe coded video transmission, or prediction of uncovered background, due to the lack of motion information. Decoder side recovery techniques work on the received data without the need for any error correction data to be transmitted by the encoder. Most algorithms reported in literature are best suited for either pure structure or pure texture areas. Although an attempt is made in [1] to classify lost blocks as structure or texture and use the suitable algorithm for each case, the issue of structure within texture blocks or vice versa is not well addressed. With the popularity of high resolution content, block sizes larger than the traditional $16 \times 16$ are becoming increasingly important for coding [2] and hence the decoder should be able to deal with large block errors.

In [3], we proposed Structure-Aware Inpainting (SAI), an algorithm controlled by segmentation and tensor voting [4], for filling arbitrary shaped regions. SAI produces good results, especially for textured regions, as it replicates the available surrounding natural texture. However, it relies on the masks generated by segmentation, which can produce visible artifacts when patching happens from an incorrect segment.

In recent years, considerable interest has been paid to sparse image modeling techniques [5]. These algorithms operate on the intuition that natural images can be decomposed into sparse combination of basic elements. Sparse modeling techniques are able to successfully capture the inherent structure in data. In this paper, we develop a novel combination of sparse modeling and structure aware inpainting to address the issues of structure and texture handling as well as higher block sizes. Firstly, we perform segmentation to identify a plausible set of segments that are related to the area to be filled. In the case of structured regions, a mask is built to select the relevant segments for modeling. This improves the sparsity prior because the unrelated segments in the rectangular neighborhood, which would otherwise contribute to the modeling, are eliminated. For regions surrounded by highly textured area, we reconstruct the structure that could be present in the missing region using tensor voting and then fill-in the texture from appropriate segments.

## 2. FRAMEWORK FOR EXTRAPOLATION

Consider a region $\mathcal{R}$ in an image consisting of known samples in area $\mathcal{A}$ and unknown samples in area $\mathcal{B}$. The process of signal extrapolation is to estimate the samples in $\mathcal{B}$ using the samples in $\mathcal{A}$. The intensities of region $\mathcal{R}$ can be interpreted as a column vector $\mathbf{f} \in \mathbb{R}^N$ , where $N$ denotes the number of samples in $\mathcal{R}$. The algorithm starts by segmenting the available neighborhood of the missing region. The spatial segmentation algorithm used in this paper is based on a multi-resolution histogram clustering method proposed by Spann and Wilson [6]. It has been selected because it is a good compromise between segmentation efficiency and complexity. We extended the approach by Spann and Wilson to account for color information [7]. For that, the components

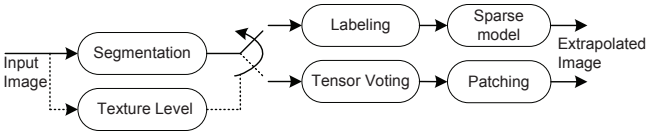


**Fig. 1**. *Block diagram of extrapolation process. Structure is reconstructed using sparse modeling and texture using Patching. Labeling produces a mask which enhances sparsity prior in modeling. Tensor Voting reconstructs structure that could be present within textured areas.*

of the color space, the given input image is represented in, are decorrelated using principal component analysis. Among the resulting color components, the one with the most discriminative power is selected for histogram clustering. It is shown in [7] that this approach significantly improves the performance of the fundamental algorithm.

Then, we estimate the characteristics of missing area by computing the texture level of surrounding samples according to [8]. It involves the computation of local extrema for each of the surrounding blocks. A local extremum is a sample which is both a local row extremum as well as a local column extremum. When the count of local extrema falls below a specified threshold, we build a sparse model of $\mathcal{R}$ and use it to estimate the samples of $\mathcal{B}$, as described in Sec. 2.1. Otherwise, we consider the missing block as containing high texture levels and determine the structure using tensor voting and perform a patch based synthesis [3], as detailed in Sec. 2.2. The entire process in depicted in a block diagram in Fig. 1.

### 2.1. Extrapolation using Sparse model

We start by building a model of the area $\mathcal{R}$ using only the samples of $\mathcal{A}$ by means of a masking vector $\mathbf{m}$ which is defined to contain a value of one at the locations of the known samples and zero at missing locations. We consider a dictionary $\mathcal{D}$ consisting of vectors $\mathbf{d}_k$. A parametric model consisting of a linear combination of vectors $\mathbf{d}_k$ is used for generating the approximation vector $\hat{\mathbf{f}}$, so that

$$\hat{\mathbf{f}} = \sum_{\forall \mathbf{d}_k \in \mathcal{K}} c_k \cdot \mathbf{d}_k, \tag{1}$$

where $k \in \mathcal{K}$ consists of the vectors chosen from $\mathcal{D}$ used for modeling and $c_k$ are the model parameters to be estimated. The estimation is done such that the approximation error between the samples at known locations of $\mathbf{f}$ and the corresponding samples produced by the model $\hat{\mathbf{f}}$,

$$E = (\mathbf{f} - \hat{\mathbf{f}})^T \cdot \mathbf{m} \cdot \mathbf{m}^T \cdot (\mathbf{f} - \hat{\mathbf{f}}), \tag{2}$$

is minimized. In [9], an isotropically decaying weighting function is defined so that the known samples in the vicinity of the unknown area get a higher importance than the samples that are far from it.

When Eq. (2) is minimized by setting the partial derivatives of $E$ w.r.t $c_k$ to zero, it leads to an underdetermined system of equations as the number of known samples is less than the total number of samples in $\mathcal{R}$. For solving this underdetermined problem, a greedy approach is taken in which the signal is approximated in terms of one additional vector from $\mathcal{D}$ per iteration [10]. In each iteration, the vector is chosen in such a way that the reduction of the weighted residual energy is maximized. After generating the parametric model $\hat{\mathbf{f}}$, the area of interest is cut out and used as an estimate for the unknown samples. In case the dictionary is composed of basis vectors of $\mathbb{R}^N$, Frequency Selective Extrapolation [11], a fast and efficient algorithm that performs sparse modeling in transform domain, can be employed.

We enhance the sparse modeling by exploiting the results of segmentation. Generally, the entire rectangular neighborhood $\mathcal{R}$ is used for the error minimization. When doing so, $c_k$s' sparsity is affected because $\mathcal{R}$ could be composed of samples of multiple segments, whereas the missing area $\mathcal{B}$ might be constituted by only few of those segments. This leads to non-homogenity and may include many more vectors from $\mathcal{D}$ into the model than actually needed. To improve the sparsity prior, we identify the segments that go through the missing area using spatial labeling of surrounding samples. Fig. 2(a) shows three different segments $S_1$, $S_2$ and $S_3$. In this example, only the segments $S_2$ and $S_3$ go through the missing area. We construct a segmentation mask $\mathbf{b}$ containing a value of one at relevant segments (labels $S_2$, $S_3$) and zero at remaining locations (label $S_1$). A new matrix $\mathbf{S}$ is formed combining the mask containing known samples and the selected segments,

$$\mathbf{S} = (\mathbf{m} \cdot \mathbf{m}^T) \cdot (\mathbf{b} \cdot \mathbf{b}^T). \tag{3}$$

The cost function for the minimization now becomes,

$$J = (\mathbf{f} - \hat{\mathbf{f}})^T \cdot \mathbf{S} \cdot (\mathbf{f} - \hat{\mathbf{f}}). \tag{4}$$

The remaining steps of dictionary vector selection and residual updation are unaltered by the proposed modification.

### 2.2. Texture synthesis by Patching

When the texture in the image is not similar to the dictionary elements, model based methods as described in Sec. 2.1 may result in blurring. One possibility to avoid such artifacts is to adapt the dictionary using the observed data. Patch based texture synthesis techniques offer a simple solution by copying natural texture from surrounding area. Here, we describe the proposed method based on [3] and discuss the cases of multiple missing textures and structure in textured areas.

Continuing the notation from Sec. 2.1, the basic process of texture replication from neighboring available data can be represented as a linear operation,

$$\hat{\mathbf{f}} = \mathbf{M} \cdot \mathbf{f}, \tag{5}$$

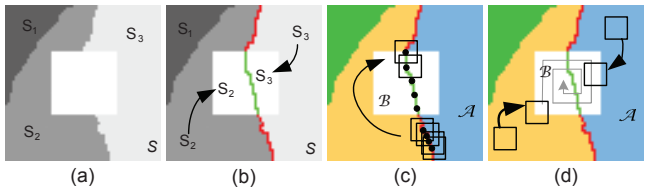


**Fig. 2**. *(a) Segmentation labels $S_1$, $S_2$ and $S_3$ with missing area shown in white. (b) Segmentation mask with the detected edges (red) and the new estimated edge by using tensor voting (green). To each resulting region in $\mathcal{B}$ the adjacent segmentation label is assigned. (c) To reconstruct the edge within $\mathcal{B}$, patches along the detected edges are used. To find the best patch, the similarities in texture and structure are measured via MSE and Euclidean distance. (d) Every position in $\mathcal{B}$ is referred to a segmentation label. To reconstruct the texture within $\mathcal{B}$, patches are taken from appropriate label positions.*

where the matrix $\mathbf{M}$ is of dimension $N \times N$. It is initialized so that a known sample is just copied from $\mathbf{f}$ to $\hat{\mathbf{f}}$. For instance, if the element at row $i$ in $\mathbf{f}$ is known, a value of one is set at the diagonal location of $\mathbf{M}$ in row $i$ and the other entries of row $i$ are set to zero. For the determination of elements of the rows corresponding to unknown samples, we use a property of texture similarity that for a generated texture to be similar to an input texture it is sufficient that all neighborhoods in the generated texture be similar to some neighborhood in the input [12].

The dominant structures, such as object contours, are important for human perception. The missing block could be composed of multiple texture patterns or a mix of structure and texture, illustrated as different segments in Fig. 2(a). In such cases, the edge between the different textures constitute important structural information. Such blocks are handled solely as structure blocks in [1]. In our proposed framework, we detect such occurrences using dominant edges between different textures in terms of segment boundaries (Fig. 2(b), red lines) and reproduce the structure within texture blocks resulting in better quality of extrapolation. Edges impinging on $\mathcal{B}$ are referred pairwise to each other and reconstructed within $\mathcal{B}$, by using tensor voting [3, 4]. The newly calculated edge is depicted in green in Fig. 2(b). This provides a restriction on $\mathbf{M}$ by zeroing out the elements that do not belong to the same segment as the sample to be estimated.

The filling process relies on the assumption that edge reconstruction is more important than texture reconstruction and hence edge filling, described in Fig. 2(c), is operated before texture filling shown in Fig. 2(d). The filling process is initialized by sampling the structure equidistantly (half patch size) to generate sample filling positions within $\mathcal{B}$ denoted as $\mathbf{v}$. The known adjacent edges in $\mathcal{A}$ are also sampled to produce source patches $\mathbf{u}$. Patches centered around the source sample positions within $\mathcal{A}$ are used to restore the missing edge in $\mathcal{B}$. Matching is conducted based on the known boundary condition, i.e. from the border of $\mathcal{B}$ inwards. The best patch is found by minimizing

$$(\mathbf{u}-\mathbf{v})^T \cdot (\mathbf{u}-\mathbf{v}) + \lambda \cdot \mathrm{dist}(\mathbf{u},\mathbf{v}), \qquad (6)$$

where $\mathrm{dist}(\mathbf{u},\mathbf{v})$ is the shortest euclidean distance between the patches $\mathbf{u}$ and $\mathbf{v}$ and $\lambda$ is the relative importance of template matching error and the proximity of patches. Texture filling is done in a helical manner starting from the border of $\mathcal{B}$ inwards as depicted in Fig. 2(d). The search area outside $\mathcal{B}$ is restricted to locations that have the same segmentation label as the considered texture location. A post-processing method [3] is introduced, to ensure a seamless transition between adjacent patches.

| Image | PSNR (dB) | | SSIM | |
|---|---|---|---|---|
| | CM[13] | Proposal | CM[13] | Proposal |
| Lena | 31.70 | 33.14 | 0.97 | 0.98 |
| Peppers | 30.74 | 30.97 | 0.93 | 0.94 |
| Baboon | 28.31 | 28.16 | 0.88 | 0.88 |

**Table 1**. *Objective results of extrapolation*

## 3. SIMULATION SETUP AND RESULTS

For evaluating the proposed algorithm, we generate input images by cutting out blocks from $512 \times 512$ test images 'Lena', 'Baboon' and 'Peppers' (Fig. 3 & 4). The input images consist of missing blocks of size $16 \times 16$, $32 \times 32$ and $64 \times 64$ samples. The missing areas are then extrapolated using the proposed algorithm. A fixed patch size of $11 \times 11$ is used to fill in the texture regions. As the dictionary elements, DFT basis functions of size $64 \times 64$ are used in case of missing blocks upto $32 \times 32$ and then increased to $128 \times 128$ for missing blocks of size $64 \times 64$. The results of extrapolation are summarized in the form of PSNR and Structural Similarity (SSIM) metrics in Tab. 1 which compares a method based on Confidence map (CM) [13] with the proposed method.

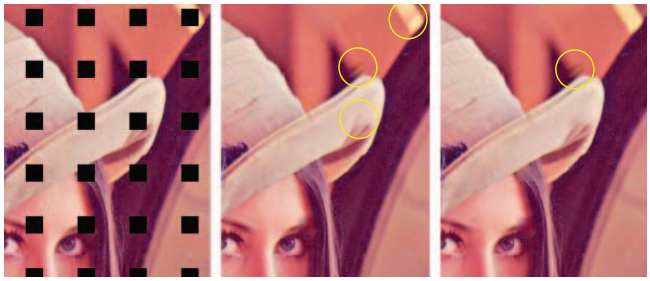

**Fig. 3**. *'Lena' image results. Left: missing blocks of size $16 \times 16$, Center: Extrapolated using sparse model, Right: Extrapolated using sparse model with segement selection. For showing the benefit due to segment selection alone, no texture patching is employed.*

(a) 'Peppers', missing blocks of size $32 \times 32$

(b) 'Baboon', missing blocks of size $64 \times 64$

**Fig. 4**. *Extrapolation results. Left: Portions of test images with missing blocks; Center: Extrapolation using Sparse model only; Right: Extrapolation using proposed method.*

For subjective evaluation of extrapolation results, portions of tested images are depicted in Fig. 3 & Fig. 4.

Compared to sparse modeling without segmentation, an improvement of upto 0.6 dB in Y-PSNR is achieved for $16 \times 16$ missing blocks because of including segment selection in sparse modeling. The effect of segment selection is evident in the top-right missing block of 'Lena' image (Fig. 3). Here, modeling using rectangular neighborhood produces a black patch in the diagonal strip. This is due to the fact that the $64 \times 64$ neighborhood of the missing block contains an unrelated black segment at the bottom-right region. The proposed algorithm excludes the unrelated segment from modeling and results in a clean strip as illustrated (Fig. 3, right). Similar reduction of artifacts due to segment selection in the test image 'Peppers' for $16 \times 16$ missing blocks was observed (not shown in Fig.). The dependancy of patching on segmentation is reduced by this framework because many structures are estimated using sparse model where segmentation is used only to build a mask for excluding unrelated areas.

The effect of texture extrapolation can be best observed in the test image 'Baboon' (Fig. 4(b)). The blurring caused by using fixed DFT basis functions, for highly textured regions, is evident in this image. Because of the loss size of $64 \times 64$, significant structural information is also missing in the input image. The recovered structures (Fig. 4(b), right) are perceptually plausible and fit seamlessly into the surrounding visual information. The resulting image appears more natural and does not contain blurring artifacts.

## 4. CONCLUSION

We presented a powerful algorithm by combining sparse modeling and structure aware texture synthesis for spatial extrapolation that can be used in variety of applications. We showed that, in case of sparse modeling, the extrapolation quality can be improved by using structural information. Textured blocks, containing multiple textures or structural information are handled using tensor voting. The extrapolated data appears natural and fit well into the surroundings. The algorithm is superior to other patch based texture synthesis algorithms, as it can reconstruct structure within textured areas even for large unknown blocks.

## 5. REFERENCES


[1] S. D. Rane, G. Sapiro, and M. Bertalmio, "Structure and Texture Filling-In of Missing Image Blocks in Wireless Transmission and Compression Applications," *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 296–303, March 2003.

[2] P. Chen, Y. Ye, M. Karczewicz, "Video coding using extended block sizes", *VCEG-AJ23*, San Diego, USA, 8-10 Oct., 2008

[3] P. Ndjiki-Nya, M. Köppel, D. Doshkov and T. Wiegand, "Automatic Structure-Aware Inpainting for Complex Image Content," *Int. Sym. on Visual Computing*, Nov 2008.

[4] G. Medioni, M. Lee, and C. Tang. "A Computational Framework for Segmentation and Grouping," *Elsevier*, New York, USA, 2000.

[5] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[6] M. Spann and R. Wilson., "A Quad-Tree Approach to Image Segmentation which Combines Statistical and Spatial Information," *Pattern Recognition.*, vol. 18, no. 3/4, pp. 257–269, 1985.

[7] P. Ndjiki-Nya, G. Simo, and T. Wiegand, "Evaluation of Color Image Segmentation Algorithms Based on Histogram Thresholding," *Int. Workshop Very Low Bitrate Video.*, 2005.

[8] K. Karu, A. K. Jain, and R. M. Bolle, "Is there any texture in the image?," *Pattern Recognition.*, vol. 29, pp. 1437-1446, 1996.

[9] K. Meisinger, and A. Kaup, "Minimizing a weighted error criterion for spatial error concealment of missing image data," *Int. Conf. Img. Proc.*, Singapore, pp. 813-816, Oct. 2004.

[10] S. Mallat, and Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[11] A. Kaup, K. Meisinger and T. Aach, "Frequency selective signal extrapolation with applications to error concealment in image communication," *Int. J. Electron. Commun.*, vol. 59, pp. 147–156, March 2005.

[12] V. Kwatra, I. Essa, A. Bobick, and N .Kwatra, "Texture optimization for example-based synthesis," *Proc. of ACM SIGGRAPH*, vol. 24, no. 3, pp. 795 - 802, July 2005.

[13] A. Criminisi, P. Perez, and K. Toyama. "Region Filling and Object Removal by Exemplar-based Image Inpainting," *IEEE Trans. Img. Proc.*, vol. 13, no. 9, pp. 1200-1212, 2004.