Sinteza
Međunarodna naučna konferencija Univerziteta Singidunum

Impact of Internet on Business Activities in Serbia and Worldwide
Uticaj Interneta na poslovanje u Srbiji i svetu

DOI: 10.15308/SINTEZA-2014-835-840

# IMAGE INPAINTING WITH DATA-ADAPTIVE SPARSITY

**Ivan V. Bajić**
School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada

**Abstract:**

Image inpainting finds numerous applications in object removal, error concealment, view synthesis, and so on. Among the existing methods, exemplar-based inpainting has been shown to achieve superior performance when filling in large areas. This paper presents a review of inpainting based on sparse representations, as a generalization of conventional exemplar-based inpainting. The importance of data-driven adaptation of the sparsity level according to the image content is emphasized. Experimental results show that incorporating data-adaptive sparsity leads to improvement in both subjective and objective inpainting performance compared to well-known exemplar-based inpainting.

**Key words:**
Image inpainting,
sparse representations,
adaptive sparsity.

## INTRODUCTION

Image inpainting [1], [2], [3] is a process of filling in parts of an image that are damaged, missing, or need to be removed, in a plausible manner, so that the resulting image maintains a natural look and feel. Some of the applications of image inpainting include:

- Object removal [4], [5], [6], where an undesired object is cut out and replaced by the data that naturally completes the image. An example is given in Fig. 1 where, at the top, an image with an undesired object (a person) in the right part is shown. The bottom image, where the undesired object has been removed, is the result of the algorithm presented in this paper.
- Error concealment [7], [8], where a part of the image is damaged due to errors in transmission. In this case, damaged parts of the image have to be filled-in based on the correctly received data.
- Disocclusion for view synthesis [9], [10], [11], where a view of the scene from a new viewpoint needs to be synthesized with the help of other views. In this case, foreground objects often occlude parts of the background that are visible from the new viewpoint. These areas need to be filled-in appropriately to generate a realistic view.

Early work on image inpainting [1], [2] was based on partial differential equation (PDE) modeling of pixel dynamics. More recently, exemplar-based methods such as [4] have become popular. In these methods, the structure and texture of the area that needs to be filled in (henceforth referred to as the "hole") is inferred by sampling from the known parts of the image. The filling proceeds step by step, from the boundary of the hole towards its interior, usually one patch at a time.

Most recently, inpainting based on sparse representation [5], [6] has emerged as an extension of early exemplar-based methods. In this approach, one assumes that the patches used to fill in the hole can be represented as sparse linear combinations of elements from a dictionary constructed from the known parts of the image.

This paper presents a review of image inpainting using sparse representation. The importance of adapting the sparsity level according to the image content is emphasized, and a simple method for doing so is described. While adaptive sparsity has been studied before in the context of image reconstruction [12], our approach is much simpler - it does not involve multilayer processing and makes use of the information already computed in the process of determining the fill order.

The paper is organized as follows. In Section II we briefly review the inpainting method of Criminisi *et al.* [4], which is considered the gold standard of examplar-based inpainting methods. In Section III we review the basics of sparsity-based inpainting and describe a simple data-adaptive approach for setting the sparsity constraint. Experimental results are presented in Section IV, followed by conclusions in Section V.

**Fig. 1.** An example of object removal by image inpainting. Top: image with a foreground object in the right part. Bottom: image with the object removed from the right part.

## EXEMPLAR-BASED INPAINTING

Among the exemplar-based methods for image inpainting, the approach of Criminisi *et al.* [4] is among the best known and most widely used. In this section, we briefly review their method and introduce the notation, which will be used throughout the paper.

When an object needs to be removed from an image, the user identifies the object by indicating the locations of its pixels in an object mask, as shown in Fig. 2 (top) for the image from Fig. 1. Alternatively, the user may indicate the object's pixels by a special color, as is common in the inpainting literature [4], [6]. When the object is cut out of the image, a hole is created, which needs to be filled in based the data from the remainder of the image.

Fig. 2 (bottom) illustrates several important concepts. The whole image is denoted $I$. The hole (also referred to as the *target region*) is denoted $\Omega$, the area with available pixels (also known as the *source region*) is denoted $\Phi$, and the boundary between $\Omega$ and $\Phi$, referred to as the *fill front*, is denoted $\delta\Omega$. The green square indicates an image patch, usually $9 \times 9$ or $11 \times 11$, centered at pixel $\mathbf{p}$, which is located on the fill front $\delta\Omega$. The patch itself is denoted $\Psi_{\mathbf{p}}$. Note that the patch covers both available pixels and missing pixels.

Vector $\mathbf{n_p}$ is a unit vector orthogonal to the fill front at point $\mathbf{p}$. Vector $\nabla I_{\mathbf{p}}^{\perp}$ is orthogonal to the image gradient at point $\mathbf{p}$, so it indicates the dominant edge direction at that point. Hence, the scalar product $\mathbf{n_p} \cdot \nabla I_{\mathbf{p}}^{\perp}$ is a measure of the extent to which edges are orthogonal to the fill front at point $\mathbf{p}$. Criminisi *et al.* [4] define a data term $D(\mathbf{p})$ that is proportional to this scalar product, as

$$D(\mathbf{p}) = \frac{\left|\mathbf{n_p} \cdot \nabla I_{\mathbf{p}}^{\perp}\right|}{\alpha}, \tag{1}$$

where $\alpha$ is the normalizing constant (typically $255$).

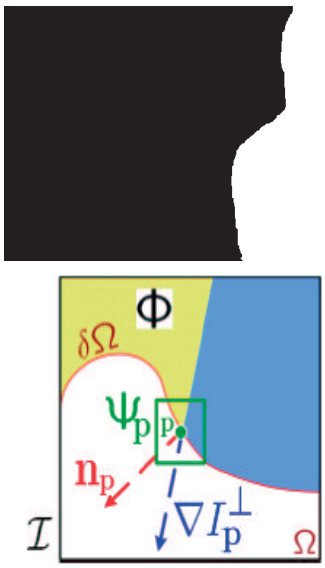


**Fig. 2.** Top: object mask for the image in Fig. 1. Bottom: adapted from [4]; illustration of the terminology used in Criminisi *et al.* inpainting.

Another important concept is the confidence term, which is defined as

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in \Psi_{\mathbf{p}} \cap (I \setminus \Omega)} C(\mathbf{q})}{\text{area}(\Psi_{\mathbf{p}})}, \tag{2}$$

where $C(\mathbf{q}) = 0$ if $\mathbf{q} \in \Omega$, and $1$ otherwise. Hence, the numerator in (2) counts how many pixels in the patch $\Psi_{\mathbf{p}}$ are available, while the whole confidence term $C(\mathbf{p})$ represents the fraction of available pixels in $\Psi_{\mathbf{p}}$.

The priority of a patch along the fill front is computed as the product of data and confidence terms, that is

$$P(\mathbf{p}) = C(\mathbf{p}) D(\mathbf{p}). \tag{3}$$

At each iteration, the patch with the maximum priority along the fill front is found, $\hat{\mathbf{p}} = \arg\max_{\mathbf{q} \in \delta\Omega} P(\mathbf{q})$ its best matching patch in the source region is identified

$$\Psi_{\hat{\mathbf{q}}} = \arg\min_{\Psi_{\mathbf{q}} \in \Phi} d\left(\Psi_{\hat{\mathbf{p}}}, \Psi_{\mathbf{q}}\right), \tag{4}$$

and the pixel values in the missing locations are transferred from $\Psi_{\hat{\mathbf{q}}}$ to $\Psi_{\hat{\mathbf{p}}}$. In (4), $d(\cdot,\cdot)$ is a measure of distance between patches, e.g., Euclidean distance. After filling in the pixel values, the confidence values of the filled-in pixels are set to $C(\hat{\mathbf{p}})$, data terms (1) are computed along the new fill front, and the procedure repeats until the entire hole is filled.

## INPAINTING BASED ON SPARSE REPRESENTATION

One of the limitations of exemplar-based inpainting in [4] is that it can only transfer existing pixel patterns from the source region into the hole. In order to allow more flexibility, one could consider linear combinations of existing pixel patterns as possible fill data. Inpainting

based on sparse representation is a formalization of this idea, where the number of terms in the linear combination is kept small.

## Sparse representation of image patches

Consider an $N \times N$ patch $\Psi_{\mathbf{p}}$ centered at $\mathbf{p}$. Let $\boldsymbol{\psi}_{\mathbf{p}}$ be a $N^2 \times 1$ column vector representing a column-wise vectorized version of $\Psi_{\mathbf{p}}$, i.e., $\boldsymbol{\psi}_{\mathbf{p}} = vec(\Psi_{\mathbf{p}})$. In the case of color images where patches are $N \times N \times 3$, different color components are stacked column-wise, so $\boldsymbol{\psi}_{\mathbf{p}}$ would be a $3N^2 \times 1$ vector. Let $\mathbf{D}$ be a $N^2 \times K$ (or $3N^2 \times K$, in case of color images) matrix, which will be referred to as the *dictionary*. Its columns, which have the same dimension as vectorized patches, will be referred to as *atoms*. Dictionary $\mathbf{D}$ can be learned from the patches source region [5], [13]. Alternatively, all patches in the entire source region can be considered as a large dictionary [6]. Sparse representation of $\boldsymbol{\psi}_{\mathbf{p}}$ in terms of the atoms in $\mathbf{D}$ can be found by solving

$$\mathbf{a}_s = \arg\min_{\mathbf{a}\in\mathbb{R}^K} \|\boldsymbol{\psi}_{\mathbf{p}} - \mathbf{D}\mathbf{a}\|_2^2, \quad \text{subject to } \|\mathbf{a}\|_0 \le \lambda, \tag{5}$$

where $\|\cdot\|_p$ stands for the $\ell_p$ norm. Vector $\mathbf{a}_s$ is referred to as the *sparse coding vector*. Solving (5) is difficult because the $\ell_0$ norm constraint is not convex. Popular workarounds include replacing the $\ell_0$ norm by the $\ell_1$ norm [14], which is convex and sparsity-promoting, or by using an iteratively reweighted $\ell_2$ approximation to the $\ell_0$ norm [15].

## Recovery based on sparse representation

Suppose $\boldsymbol{\psi}_{\mathbf{p}}$ represents a patch on the fill front, such that some of its pixels are missing. Let $\bar{\boldsymbol{\psi}}_{\mathbf{p}}$ be the column vector of dimension $M \times 1$ where $M < N^2$ ($M < 3N^2$ for color images), which is obtained from $\boldsymbol{\psi}_{\mathbf{p}}$ by removing the elements corresponding to the missing pixels. Analogously, let $\bar{\bar{\mathbf{D}}}$ be the truncated dictionary, obtained from $\mathbf{D}$ by removing the rows corresponding to the missing pixels in $\boldsymbol{\psi}_{\mathbf{p}}$. Then the missing pixels in $\boldsymbol{\psi}_{\mathbf{p}}$ can be approximately recovered by finding a sparse representation of $\bar{\boldsymbol{\psi}}_{\mathbf{p}}$ in terms of $\bar{\bar{\mathbf{D}}}$,

$$\mathbf{a}_s = \arg\min_{\mathbf{a}\in\mathbb{R}^K} \|\bar{\boldsymbol{\psi}}_{\mathbf{p}} - \bar{\bar{\mathbf{D}}}\mathbf{a}\|_2^2, \quad \text{subject to } \|\mathbf{a}\|_0 \le \lambda \tag{6}$$

then using $\mathbf{a}_s$ to recover the full patch $\boldsymbol{\psi}_{\mathbf{p}}$ from $\mathbf{D}$,

$$\boldsymbol{\psi}_{\mathbf{p}} = \mathbf{D}\mathbf{a}_s. \tag{7}$$

Note that if $\mathbf{D}$ and $\bar{\bar{\mathbf{D}}}$ are normalized to contain unit column vectors, as would normally be the case when using fast sparse solvers [13], then the elements of $\mathbf{a}_s$ obtained from (6) need to be scaled appropriately before computing (7). Also note that when $\mathbf{D}$ contains all the patches in the source region and $\lambda = 1$, sparse recovery is equivalent to the exemplar-based inpainting described in Section II, when $d(\cdot,\cdot)$ in (4) is the squared Euclidean distance. Hence, inpainting based on sparse recovery is a generalization of exemplar-based inpainting.

## Adapting the sparsity level

Fig. 3 shows an image inpainted using the exemplar-based approach described in Section II, when the source region is the entire image minus the hole. While the grass field in the lower right part is inpainted reasonably well, an artifact is created in the smoother region of the sky above the tree line. Similar artifacts have been observed by the authors in [6]. The reason for such behavior is that low texture in smooth regions does not provide sufficient discrimination of matching patches in the source region, potentially leading to false matches and creating artifacts such as those shown in Fig. 3.

On the other hand, recovery based on sparse representation inherently possesses smoothing capabilities. Since the recovered patch is a weighted average of selected dictionary members, the higher the number of non-zero terms in the sparse coding vector $\mathbf{a}_s$, the smoother the resulting patch can be expected to be. It would therefore seem beneficial to relate the sparsity constraint $\lambda$ to the desired smoothness of the reconstructed patch. In order to do this, one can make use of the data term $D(\mathbf{p})$, which measures the strength of the edges incident on the fill front. The higher $D(\mathbf{p})$ is, the lower $\lambda$ should be.

In our implementation, we have used the following approach to adapt $\lambda$:

$$\lambda(\mathbf{p}) = \left\lfloor 1 + \alpha \cdot \mathrm{e}^{-(\beta \cdot D(\mathbf{p}))^2} \right\rfloor, \tag{8}$$

where $\alpha$ and $\beta$ are constants and $\lfloor x \rfloor$ represents the largest integer no greater than $x$. Suitable values for $\alpha$ and $\beta$ were empirically found to be $\alpha = 3$ and $\beta = 40$. As $D(\mathbf{p})$ increases, meaning that the strength of edges incident on the fill front increases, $\lambda(\mathbf{p})$ reduces to $1$, that is, the inpainting method becomes exemplar-based. At the other extreme, if $D(\mathbf{p}) = 0$, $\lambda(\mathbf{p})$ becomes $1 + \alpha$, which is $4$ with our parameter settings. Hence, up to $4$ dictionary elements will be selected for sparse recovery in smooth areas.

**Fig. 3.** Inpainting of the image in Fig. 1 using the method of Criminisi *et al.* [4]. Note the artifact above the tree line in the right part of the image.

In our implementation, the sparse coding problem (6) is solved via Matching Pursuit (MP) [16]. Although generally suboptimal, it leads to reasonably good results (e.g., Fig. 1 bottom) and it can be computed in at most $\lambda(\mathbf{p})$ iterations. The first iteration amounts to finding the column of $\bar{\bar{\mathbf{D}}}$ that is most correlated with $\bar{\boldsymbol{\psi}}_{\mathbf{p}}$, which is essentially

the same as solving (4) when $d(\cdot,\cdot)$ is the squared Euclidean distance. Subsequent iterations perform the same procedure using the current approximation error instead of $\bar{\psi}_{\mathbf{p}}$ itself. The source region is set to be the neighborhood of the hole dilated by a square structuring element of dimension $80$. The dictionary is taken to contain all patches in the source region; in other words, no dictionary learning is employed.

It is worthy of noting that [6] employed a similar method for inpainting based on sparse representation, but with the following important differences. First, the $\ell_0$ norm was replaced by the $\ell_1$ norm in (6), which necessitated using a different sparse coding algorithm. Second, the size of the source region was not clearly specified; it was mentioned that the entire image minus the hole could be used as the source region. Finally, and perhaps most importantly, sparsity adaptation was not considered.

## EXPERIMENTS

We compare the presented adaptive-sparsity method with two versions of the Criminisi *et al.* [4] approach. The first is the "default" version, where the source region is the entire image minus the hole, and the other is the "restricted" version, where the source region is the same as in the adaptive-sparsity method - the neighborhood of the hole dilated by a square structuring element of dimension $80$.

Two methodologies have emerged for testing image inpainting methods. In one approach, a natural image is taken, and an object from this image is selected for removal [4], [6]. This approach has the advantage of mimicking practical applications of object removal, however, the downside is that the performance cannot be judged objectively, because it is not known what really lies behind the object that needs to be removed. There is no objective ground truth, so the results are only judged subjectively. The other approach is to deliberately insert an object into an image and then try to remove it. This approach was taken in [5] by adding text to an image, and then removing it. Although somewhat artificial, the advantage of this approach is that a well-defined ground truth exists, so that both objective and subjective assessment of the inpainting method is possible. In this work, we take the latter approach. However, instead of adding text, we add a large object (e.g., the person in Fig. 1 top), which leads to a more challenging inpainting problem.


**Fig. 4.** Some visual results.

Specifically, $12$ images were selected from the image database [17]. The database contains images of various resolution. The images selected for experiments had a 4:3 aspect ratio and were resized to $480 \times 360$, without changing the aspect ratio. For each image, the object was inserted once in the left part and once in the right part of the image, giving a total of $24$ test images. Several examples are shown in Fig. 4. The first column shows the original image, followed, respectively, by the image with the object inserted, and the results of default Criminisi inpainting, restricted Criminisi inpainting, and the adaptive-sparsity method.

It can be seen from Fig. 4 that all three inpainting methods provide considerable level of realism in the inpainted images. At first look, it is often not immediately obvious whether anything is wrong with these images. A closer examination, however, reveals the presence of various artifacts, most notably in the default Criminisi result, but also, to a lesser extent, in the restricted Criminisi result and in the images produced by the adaptive-sparsity method. It can also be seen that some features of the original image cannot be recovered by inpainting. For example, in the bottom row, the person standing on the street in the original image (first column), who is completely obstructed by the inserted object (second column), cannot be recovered, since there is no information about the presence of this person in the remainder of the image.

Next we turn to objective evaluation. For this purpose, we utilize the Peak Signal to Noise Ratio ($PSNR$) in dB, defined as

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}, \tag{9}$$

where $MSE$ is the mean squared error between the luminance components of pixels from the original image and pixels in the inpainted image. Results are shown in Table I. The three digits in the image name identify the image index in the database [17] and the trailing letter (L/R) indicates whether the object was inserted in the left or right part of the image. For each image, the best result is highlighted in bold typeface. When the difference between the top two $PSNR$ values is less than 0.05 dB, both are highlighted, since such difference is considered too small for meaningful distinction.

As seen in the table, each of the three methods sometimes achieves the top result. Specifically, the default Criminisi method achieved the top score 5 times, the restricted Criminisi approach was the highest-scoring 4 times, and the adaptive-sparsity method 19 times. However, on average, restricted Criminisi approach is better than the default Criminisi approach by about 0.6 dB, while the adaptive-sparsity method achieved a 0.5 dB advantage over the restricted Criminisi approach, and 1.1 dB over the default one. These results, together with the subjective results in Fig. 4, indicate that it is advantageous to restrict the source region to the neighborhood of the hole, and further gains can be achieved by averaging the patches in the source region in a content-adaptive manner, as is done in the adaptive-sparsity method.

Table 1. Objective inpainting results

| Image | Luminance ***PSNR*** (dB) of inpainted holes | | |
|---|---|---|---|
| | Default Criminisi | Restrict. Criminisi | Adaptive sparsity |
| i009L | 16.40 | **17.53** | 17.34 |
| i032L | 23.16 | 26.15 | **27.85** |
| i038L | **22.34** | 21.71 | 21.59 |
| i052L | 15.40 | **15.86** | **15.83** |
| i059L | 18.94 | **21.27** | **21.28** |
| i088L | **21.62** | 20.11 | 20.36 |
| i089L | 18.65 | 18.44 | **19.00** |
| i094L | 18.96 | 20.93 | **21.07** |
| i109L | 18.54 | 17.76 | **19.03** |
| i116L | 18.48 | 19.20 | **19.86** |
| i155L | 16.91 | 18.02 | **18.53** |
| i255L | 14.75 | 15.95 | **16.17** |
| i009R | **16.51** | 15.45 | 15.81 |
| i032R | 24.41 | 27.10 | **27.76** |
| i038R | 22.79 | 22.38 | **22.85** |
| i052R | 17.31 | 15.79 | **17.51** |
| i059R | 18.32 | 18.28 | **18.39** |
| i088R | 21.35 | 23.29 | **23.84** |
| i089R | 18.91 | 21.45 | **21.74** |
| i094R | 19.99 | 20.07 | **21.08** |
| i109R | **18.12** | 17.31 | 17.39 |
| i116R | 19.88 | 19.52 | **20.53** |
| i155R | 16.65 | 17.96 | **18.40** |
| i255R | 18.86 | **20.84** | **20.87** |
| **Avg.** | 19.05 | 19.68 | **20.17** |

## CONCLUSIONS

We have reviewed several approaches for image inpainting and presented an inpainting method based on sparse representation, where the sparsity constraint is adaptively adjusted according to the edge content incident on the fill front. Adaptation is simple and effective, making use of the data already computed in selecting the fill order. Results indicate reasonable improvements in both subjective and objective quality of inpainted images.

## REFERENCES

[1] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. "Image inpainting," *Proc. SIGGRAPH'00*, New Orleans, USA, July 2000.

[2] G. Sapiro, "Image inpainting," *SIAM News*, vol. 35, no. 4, May 2002.

[3] C. Guillemot, and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 127-144, Jan. 2014.

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

[5] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans Image Processing*, vol.17, no.1, pp. 53-69, Jan. 2008.

[6] B. Shen, W. Hu, Y. Zhang, Y.-J. Zhang, "Image inpainting via sparse representation," *Proc. IEEE ICASSP'09*, pp. 697-700, Apr. 2009.

[7] G.-L. Wu, C.-Y. Chen, and S.-Y. Chien, "Algorithm and architecture design of image inpainting engine for video error concealment applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 6, pp. 792-803, June 2011.

[8] M. Ebdelli, O. Le Meur, and C. Guillemot, "Loss concealment based on video inpainting for robust video communication," *Proc. EUSIPCO'12*, pp. 1910-1914, Aug. 2012.

[9] I. Daribo and B. Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," *Proc. IEEE MMSP10*, pp. 167-170, Oct. 2010.

[10] I. Ahn and C. Kim, "Depth-based disocclusion filling for virtual view synthesis," *Proc. IEEE ICME'12*, pp. 109-114, Jul. 2012.

[11] S. Reel, G. Cheung, P. Wong, and L. S. Dooley, "Joint texture-depth pixel inpainting of disocclusion holes in virtual view synthesis," *Proc. APSIPA Annual Summit and Conference 2013*, Oct.-Nov. 2013.

[12] O. G. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part II: adaptive algorithms," *IEEE Trans. Image Processing*, vol. 15, no. 3, pp. 555-571, Mar. 2006.

[13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Machine Learning Research*, vol. 11, pp. 19-60, Jan. 2010.

[14] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal $\ell_1$-norm solution is also the sparsest solution," *Comm. Pure Appl. Math.*, vol. 59, no. 6, pp. 797-829, June 2006.

[15] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Comm. Pure Appl. Math.*, vol. 63, no. 1, pp. 1-38, Jan. 2010.

[16] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397-3415, Dec. 1993.

[17] http://people.csail.mit.edu/tjudd/SaliencyBenchmark/