doi:10.2498/cit.1000677

Missing Data Correction in Still Images Using Multi-Resolution Analysis

Blerim Mustafa[1], Vladimir Trajkovik[2] and Danco Davcev[2]

[1]Makedonski Telekomunikacii, Skopje, Macedonia
[2]Faculty of Electrical Engineering, Skopje, Macedonia

This paper proposes an improvement to the texture generation-based image in-painting algorithm, using multi-resolution analysis. Instead of optimizing the size of the neighborhood window of the synthesized pixel and the size of the test image, the approach described in this paper applies the same (minimal) window and test image size to different sub-bands of the discrete wavelet transformed (DWT) image. This way the execution complexity is kept lowest possible, while still obtaining the same visual quality as by applying the optimal parameters in the spatial domain.

Keywords: image processing, restoration, constrained synthesis, non-parametric and non-stationary Markovian models, multi-resolution, discrete wavelet transform.

1. Introduction

Image in-painting or missing data correction in still images (as well as in video sequences) is an interesting new point in a multitude of applications since defects are common in large non-electronic archives. Another interesting application of the automatic in-painting would be the concealment of selected objects in retouching of images or other special effects.

Several approaches in automatic in-painting are distinguishable:

- Correction using reshaping the Fourier spectrum of the region of interest according to a reference spectrum (*HIRANI et al. 1996*)
- Correction using Partial Differential Equations (*BERTALMI et al. 2000*)
- Correction using multilevel median filters (*KOKARAM et al. 1995*)
- Correction by texture generation (non-parametric sampling) (*ASHIKHMIN 2001, EFROS et al. 2001, EFROS et al. 1999, BORNARD et al. 2003, IGEHY et al. 1997*)

The last category is the one that inspired our work. The next Section describes the basics of this approach and the Section 3 describes how it can be improved using multi-resolution analysis. Section 4 gives an overview of the results, using the improvement we propose in Section 3.

2. Correction Using Texture Generation

The algorithm developed in *EFROS et al. 1999* “grows” texture, pixel by pixel, outwards from an initial seed. A single pixel p is chosen as unit of synthesis so that the model can capture as much high frequency information as possible. All previously synthesized pixels in a square window around p (weighted to emphasize local structure) are used as the context. To precede with synthesis, the probability tables for the distribution of p are required, given all possible contexts. However, constructing them explicitly is out of the question. Instead, for each new context, the sample image is queried and the distribution of p is constructed as a histogram of all possible values that have occurred in the sample image as shown in Figure 1. The non-parametric sampling technique, although simple, is very powerful at capturing statistical processes for which a good model hasn’t been found.

The texture is modeled as a Markov Random Field (MRF). That is, we assume that the probability distribution of brightness values for a pixel given the brightness values of its spatial

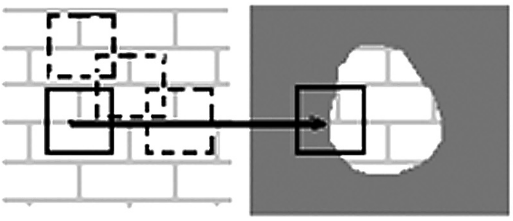

Fig. 1. Given a sample texture image (left), a new image is being synthesized, one pixel at a time (right). To synthesize a pixel, the algorithm first finds all neighborhoods in the sample image (boxes on the left) that are similar to the pixel's neighborhood (box on the right) and then chooses one neighborhood and takes its center to be the newly synthesized pixel.

neighborhood is independent from the rest of the image. The neighborhood of a pixel is modeled as a square window around that pixel. The size of the window and the size of the sample image are free parameters that specify how stochastic the user believes the texture to be. More specifically, if the texture is presumed to be mainly regular at high spatial frequencies and mainly stochastic at low spatial frequencies, the size of the window should be on the scale of the smallest regular feature, and the size of the test image should be on the scale of the largest regular feature.

Let I be an image that is being synthesized from a texture sample image $I_{smp} \subset I_{real}$ where I_{real} is the real infinite texture. Let $p \in I$ be a pixel and let $\omega(p) \subset I$ be a square image patch of width ω centered at p. Let $d(\omega 1, \omega 2)$ denote some perceptual distance between two patches. If all the pixels in I except for p are known, p can be synthesized using the conditional probability distribution $P(p|\omega(p))$ and then sample from it. A modification of the nearest neighbor is used: the closest match $\omega_{best}(s) = \arg\min_{\omega(s)} d(\omega(p), \omega(s))$ is taken for synthesizing. The distance d is usually calculated as a sum of the squared differences metric, weighted by a Gaussian kernel:

$$d(\omega(p), \omega(s)) = \frac{\sum\limits_{i,j\in\omega} \left(l_{i,j}(p) - l_{i,j}(p)\right)^2 G_{i,j} b_{i,j}(p)}{\sum\limits_{i,j\in\omega} G_{i,j} b_{i,j}(p)},$$

where $G_{i,j}$ is the Gaussian mask, $b_{i,j}(p)$ is a Boolean mask indicating whether the pixel(i,j) exists (is synthesized yet.), and $l_{i,j}$ is the luminance of the pixel(i,j).

BORNARD et al. 2003 used this scheme for missing data correction in still images. The difference between the texture generation and the natural pictures correction is that the blotches may occur in less-stationary texture or contour segments. The neighborhood of the synthesized pixel is used as a sample image. *BORNARD et al. 2003* also add adaptive sample image (the sample image grows as we go "deeper" into the blotch) and partial similarity computation to speed up the execution.

However, the size of the window and the size of the sample image are still left for the user to choose. Further, cases exist where these parameters need to be set high, which adds up to the execution time proportionally. The next chapter shows how using multi-resolution analysis helps us overcome the problem of the estimation of these free parameters, in fact, helps us to keep them low, and therefore shortens the execution time of the algorithm.

3. Correction Using Multi-Resolution

As stated in the previous section, the size of the window and the size of the sample image are free parameters that specify how stochastic the user believes the texture to be. The estimation of these parameters can be done using frequency feature extraction (using FFT, Wavelet or some other transform), or it can be done using texture classification, segmentation and comparison (*UNSER 1995, HAVLICEK et al. 2001*).

However, the approach we are proposing is pretty much the opposite: instead of using frequency analysis to get the frequency distribution of the image segment, and there from estimate the parameters from the dominant frequencies, we can analyze the image into frequency bands (using DWT) (*STRANG et al. 1996*) (Figure 2), and apply the missing data corrections in each sub-band. The highest frequency bands contain the spatial features occurring in around 2 to 4 pixels, and therefore windows of size 5 and sample images of size 9 are sufficient for corrections. Since they are decimated twice, even windows of size 3 and sample images of size

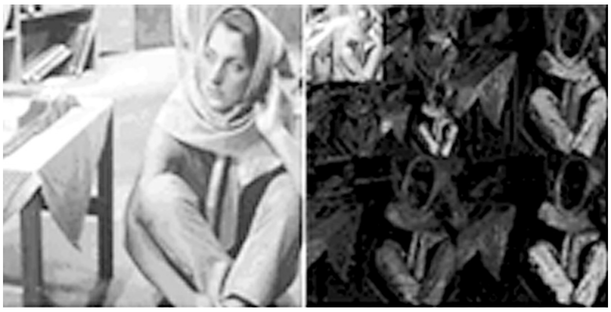

Fig. 2. The original image (left) and the 2-level wavelet transformed image (right).

5 are sufficient for corrections. The next three bands contain the variations occurring in 4 to 8 pixels, but are decimated by 4, so windows of size 3 and sample images of size 5 are sufficient for corrections here, too. The same goes for all the remaining sub-bands. The lowest sub-band contains only the contours of the image, so the same low parameters can be applied.

So, instead of estimating the parameters, we apply the smallest possible ones (and therefore "cheapest" from the execution point of view) to the wavelet-transformed image. The blotch is spatially distributed through the sub-bands (Figure 3), so we practically have the same number of blotch-pixels for corrections, but the correction is done using the smallest window and sample image sizes.

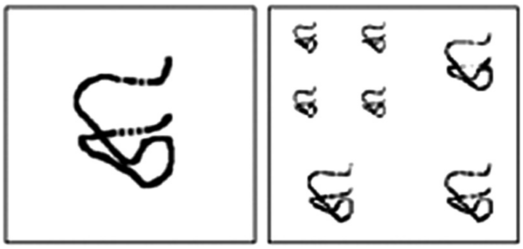

Fig. 3. The original blotch mask (left) and the spatially distributed one (right) for correcting 2-level wavelet-transformed images.

Figure 4 shows the schematic diagram of the proposed approach. Instead of estimating the parameters (visually or automatically), the image is wavelet-transformed and the blotch mask is spatially distributed according to the level of the transform. Then the transformed image is in-painted using small window and test image size, and the resulting image is inverse transformed.

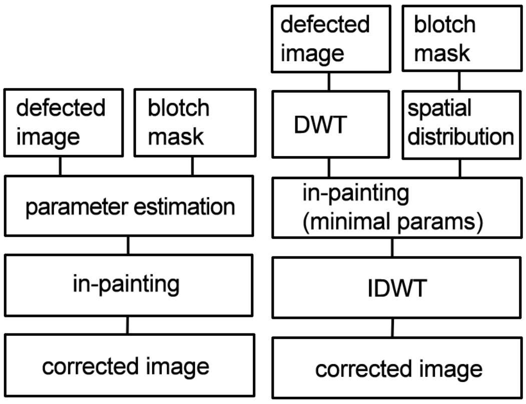

Fig. 4. Left – schematic block diagram of image in-painting by texture generation and right – block diagram
of our multi-resolution approach.

4. Results and Demonstration

The algorithm presented in *EFROS et al. 1999* and *BORNARD et al. 2003* has the computational complexity of:

$$CC = \sum_{i=1}^{n} w_i^2 s_i^2,$$

where w_i is the size of the window around the pixel i, s_i is the size of the test image for the pixel i and n is the number of blotch-pixels. For optimal correction the free parameters w_i and s_i should vary depending on the spatial distribution of frequency features around the pixel i. So, the CPU load for an optimal correction depends on the number of blotched pixels and on the frequency features surrounding those pixels.

The approach in image in-painting presented in this paper does not optimize the in-painting parameters based on the spatial distribution of frequency features in the image, but instead, applies the same (minimal) parameters on different frequency bands. So, the computational complexity would be:

$$CC_{mr} = n w_{\min}^2 s_{\min}^2$$

Therefore the CPU load for an optimal reconstruction depends only on the number of the pixels that are being synthesized, i.e. on the size of the blotches. Keeping the minimal CPU-load, this approach achieves practically the same visual quality as the previous approaches on the optimal complexity level.

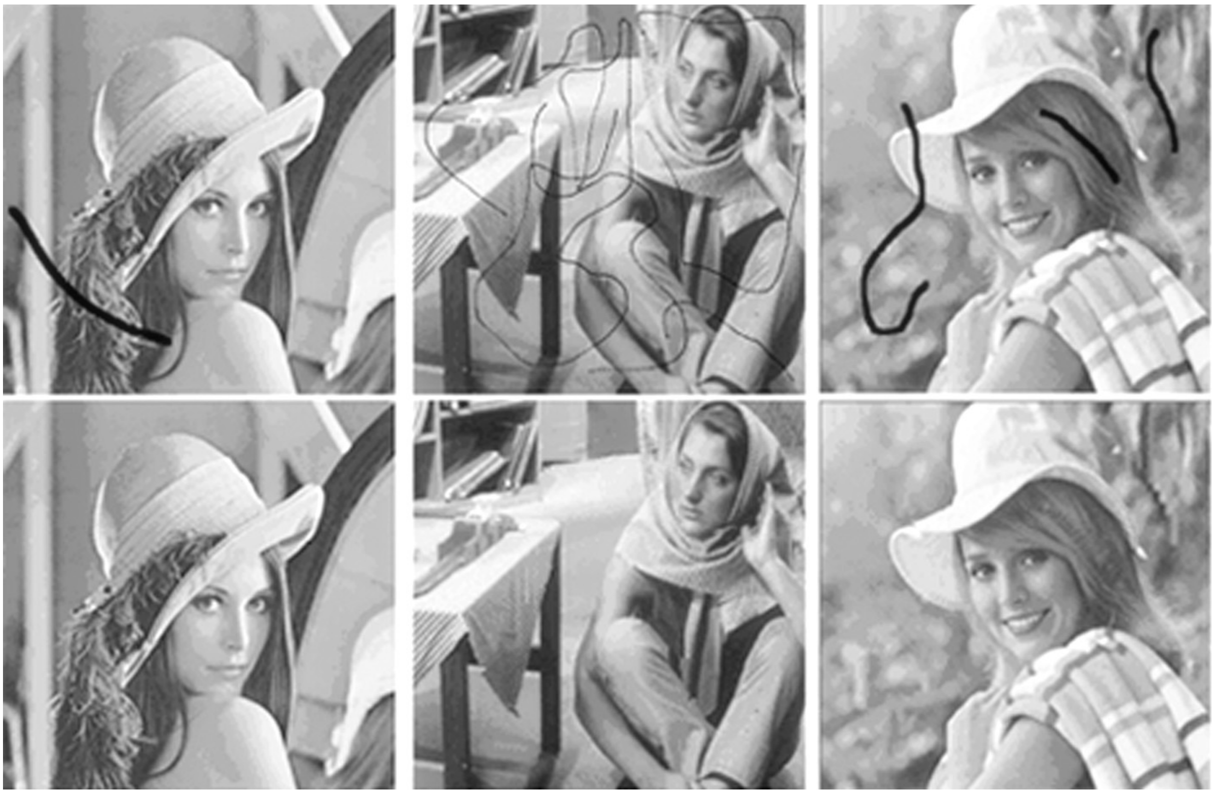

Fig. 5. Top-images with blotch masks, bottom – restored images.

Figure 5 shows three examples of the approach presented. The first and the last ones contain a large blotch going through different texture and contour regions. The second one is a randomly vandalized image, with blotches in almost all the regions. Using the same window and sample image size, the image is recovered in all the regions, despite the differences in the frequency features.

5. Conclusion

In this paper, an improved texture-generation-based image in-painting algorithm has been proposed in order to obtain a scheme for overcoming the problem of estimating free parameters (the size of the window and the size of the test image), using multi-resolution.

The visual results achieved by this multi-resolution approach where minimal parameters are used (lowest CPU Load) are practically the same as those achieved by the conventional spatial domain approach, i.e. using adjusted parameters.

References

[1] A. A. EFROS, T. K. LEUNG, Texture Synthesis by Non-Parametric Sampling. Presented in the *Proceedings of the 1999 IEEE International Conference on Computer Vision* (1999), pp. 1033–1038. Corfu, Greece.

[2] A. A. EFROS, W. T. FREEMAN, Image Quilting for Texture Synthesis and Transfer. Presented in the *Proceedings of ACM SIGGRAPH* (2001), pp. 341–346. Los Angeles, USA.

[3] A. C. KOKARAM, R. D. MORRIS, W. J. FITZGERALD, P. J. RAYNER, Interpolation of Missing Data in Image Sequences. *IEEE Transactions on Image Processing* 4 (1995), pp. 1509–1519.

[4] A. N. HIRANI, T. TOTSUKA, Combining Frequency and Spatial Domain Information for Fast Interactive Image Noise Removal. Presented in the *Proceedings of ACM SIGGRAPH* (1996), pp. 269–276. New Orleans, USA.

[5] G. STRANG, T. NGUYEN, *Wavelets and Filter Banks.* Wellesley-Cambridge Press, Wellesley, 1996.

[6] H. IGEHY, L. PEREIRA, Image Replacement Through Texture Synthesis. Presented in the *Proceedings of the 1997 IEEE International Conference on Image Processing*(1997), Santa Barbara, USA.

[7] J. P. HAVLICEK, P. C. TAY, Determination of the Number of Texture Segments Using Wavelets. 16th Conference on Applied Mathematics, Univ. of Central Oklahoma. *Electronic Journal of Differential Equations* (2001), pp. 61–70.

[8] M. ASHIKHMIN, Synthesizing Natural Textures. Presented in the *Proceedings of the 2001 ACM Symposium on Interactive 3D* (2001), pp. 217–226. Research Triangle Park, USA.

[9] M. BERTALMIO, G. SAPIRO, V. CASELLES, C. BALLESTER, Image In-painting. Presented in the *Proceedings of ACM SIGGRAPH* (2000), pp. 417–424. New Orleans, USA.

[10] M. UNSER, Texture Classification Using Wavelet Frames. *IEEE Transactions on Image Processing* 4 (1995), pp. 1549–1560.

[11] R. BORNARD, E. LECAN, L. LABORELLI, J. H. CHENOT, Missing Data Correction in Still Images and Image Sequences. Presented in the *Proceedings of ACM Multimedia* (2003), pp. 355–361. Juan-les-Pins, France.

Received: April, 2005
Accepted: December, 2006

Contact addresses:

Blerim Mustafa
Makedonski Telekomunikacii
Orce Nikolov bb
Skopje
Macedonia
e-mail: blerim.mustafa@mt.com.mk

Vladimir Trajkovik
Faculty of Electrical Engineering
Karpos II bb 1000
Skopje,
Macedonia
e-mail: trvlado@cerera.etf.ukim.edu.mk

Danco Davcev
Faculty of Electrical Engineering
Karpos II bb 1000
Skopje,
Macedonia
e-mail: etfdav@etf.ukim.edu.mk

BLERIM MUSTAFA is a PhD student at the Department of Computer Science of the Faculty of Electrical Engineering and Information Technologies in Skopje. His major fields of interest are signal, image, video and 3D data processing, as well as various fields of Artificial Intelligence.

DR. VLADIMIR TRAJKOVIK is an assistant professor at the Department of Computer Science, of the Faculty of Electrical Engineering and Information Technologies, “Sts. Cyril and Methodius” University in Skopje, Republic of Macedonia. He is author of over 40 papers published at international conferences. He teaches Algorithms and Data Structures, Distributed Computer Sciences, and Expert Systems. His research interests are: Web-based Systems, Intelligent Systems and Distance Education Systems.

DR. DANCO P. DAVCEV received his “Doctor – Ingenieur” degree in Computer Science from the University of ORSAY (Paris, France) in 1975 and PhD degree in Informatics from the University of Belgrade in 1981. He is currently a professor, head of the Computer Science and Informatics Department and head of the EU Open and Distance Learning Center at the University “Sts. Cyril and Methodius”, Skopje (Macedonia). He has more than 150 research papers presented at international conferences or published in international journals in Computer Science and Informatics. His research interests include Multimedia Databases and Educational Systems, Spatio-temporal data and 3D objects, Intelligent Information Systems and UML, Distance Learning Systems, Distributed and Multi-agent Systems, Wireless Sensor Systems.