ADAPTIVE 2D-AR FRAMEWORK FOR TEXTURE COMPLETION

F. Racape[⋆] *M. Köppel*[⋆†] *D. Doshkov*[⋆] *P. Ndjiki-Nya*[⋆]

[⋆] Fraunhofer Institute for Telecommunications Heinrich Hertz Institute (HHI)
Image Processing Department,
Einsteinufer 37. 10587 Berlin, Germany
[†] Berlin Institute of Technology,
Department of Telecommunication Systems, School of Electrical Engineering and Computer Science,
Einsteinufer 17. 10587 Berlin, Germany

ABSTRACT

Texture extrapolation techniques enable to fill large holes of missing information. Many applications can be targeted such as image and video coding, channel block losses, object removal, filling of 3D disocclusions etc. For more than two decades, many approaches have been developed, even though each contains pros and cons which force to choose the best compromise for the targeted application. In this paper, we propose to continue exploring and improving a popular parametric completion method using the autoregressive (AR) model. In this framework, the training area is automatically optimized. A consistency criterion also enables to assess and regularize the model. Moreover, a post-processing step enables to remove the remaining seam artefacts. A comparison with the state-of-the-art is provided for both subjective quality and complexity which remains a major constraint for texture completion.

Index Terms— Texture completion, parametric method, autoregressive model.

1. INTRODUCTION

Texture extrapolation appears in two overlapping domains: texture synthesis and inpainting. Texture synthesis algorithms [1] are dedicated to create a large texture from a short input image. Inpainting methods [2] refer to the extrapolation of signal inside a removed region, propagating structures. In this paper, the term texture completion is used as we aim at seamlessly filling textures, which includes both synthesis and inpainting. Texture synthesis algorithms are mostly divided into two main categories of algorithms: parametric and non-parametric approaches. The first approximates the Probability Density Function (PDF) of the input sample with a compact model [3, 4, 5, 6, 7, 8]. The most common techniques rely on the Autoregressive (AR), the Moving Average (MA) and the Autoregressive Moving Average (ARMA) models. The second [9, 10, 11, 12, 13] often build the output by searching for a best match region in the input and copying it with some different methods and region shapes depending on the method. One other category, more dedicated to inpainting, uses a diffusion process based on non-linear Partial Differential Equations (PDE) [14, 15, 16, 17].

This work was carried out during the tenure of an ERCIM “Alain Bensoussan” Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 246016.

Additionally, this work was funded in part by the German Research Foundation [Deutsche Forschungsgemeinschaft (DFG)], under the grant WI 2032/4-1.

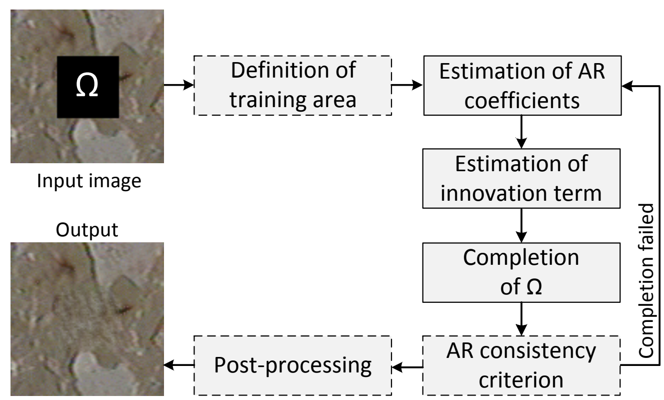


Fig. 1. Block diagram of the framework: three AR classical (solid blocks) and three improvement (dashed blocks) steps.

With respect to the AR technique, Chellappa et al. [6] used a 2D non causal autoregressive (NCAR) model to synthesize different natural texture samples sized 64×64 with several neighbour sets and parameters. In the work of Deguchi [18], blocks with similar AR parameters were merged iteratively. The work of Tugnait [7] investigated the suitability of 2D NCAR models with asymmetric support for completion of 128×128 textures. The AR model has also been used in image and video reconstruction applications. In the work of Szummer [8], the temporal textures were modeled

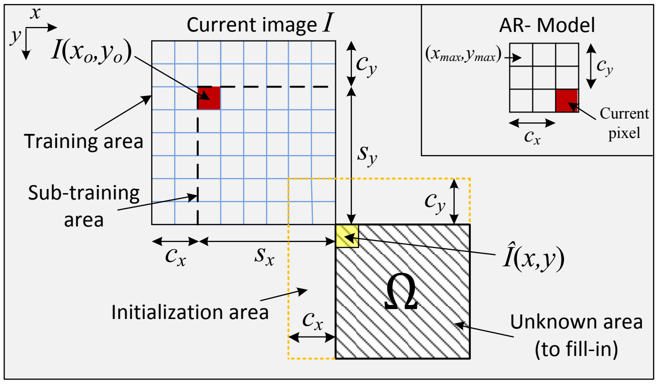


Fig. 2. AR parameter definition. Basic example of implementation with a square training area.

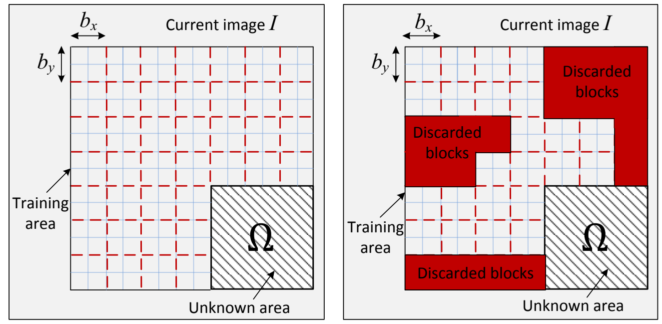


Fig. 3. Stationarity criterion on a training area adjacent to the unknown area (Ω). (left) The training region is divided into blocks of size $b_x \times b_y$ and (right). Unstationary blocks are discarded from the training area.

by a spatio-temporal AR model used to synthesize video sequences, using large causal neighbourhoods containing over 1000 parameters.

In previous AR works, important modules are not taken into account, e.g. an adequate quality assessment control, an appropriate training decision procedure, tackling possible texture completion failure. Therefore, our contribution is to provide a complete AR-modelling framework that can be integrated in several texture completion applications. An adaptive training area is designed to contain relevant and stationary texture only. Then, a consistency criterion is used to detect erroneous synthesis results. In case of errors the texture synthesis is repeated using an additional regularization term which prevents overfitting the AR model. A post processing step based on Poisson cloning [19] finally enables the removal of visible seams.

2. COMPLETION FRAMEWORK

The input image in figure 1 presents our basic use case with a picture I, in which there is a missing region Ω. The texture completion framework contains four basic steps: the definition of the training area, the estimation of AR coefficients, the estimation of the innovation term and the completion of the missing region Ω. We propose to automate and improve the classical AR scheme by adding the optional steps shown in dashed blocks in Fig. 1. They automate the scheme by self-designing a stationary training area and validating the synthesis by means of a consistency criterion. A post-processing step is finally used to remove potential localised artefacts.

In the next section, the AR model computation and filling process are described, before presenting the contributions.

3. IMPROVEMENTS TO AR COMPLETION

The AR technique considers textured images as a Markov Random Field (MRF), a single pixel is conditioned upon a spatial neighbourhood of samples. Technically, it fills region Ω one pixel at a time using a linear combination (AR parameters) of its causal neighbours plus an additive innovation term. The AR model can thus be expressed as

$$\hat{I}(x,y) = \sum_{j=y_{min}}^{y_{max}} \sum_{j=x_{min}}^{x_{max}} \alpha_{i,j} I(x-i, y-j) + \epsilon(x,y) \quad (1)$$

with $(i,j) \neq (0,0)$. $\hat{I}(x,y)$ represents the completed sample at location (x,y) in the current image I. (i,j) determine the known spatial neighbourhood values. y_{min}, y_{max}, x_{min} and x_{max} are constants that characterize the model order (cf. Fig. 2) and $\alpha_{i,j}$ correspond to the prediction coefficients. In this work we utilize a causal AR neighbourhood model as depicted in Fig. 2 (AR-Model). The function $\epsilon(x,y)$ is a white noise process with zero mean and variance σ^2. $\epsilon(x,y) \sim N(0,\sigma^2)$ and denotes the innovation signal which drives the AR model. Due to the fact that the (additive) Gaussian noise provides a good noise approximation of many real-world applications, $\epsilon(x,y)$ is typically represented by white Gaussian noise.

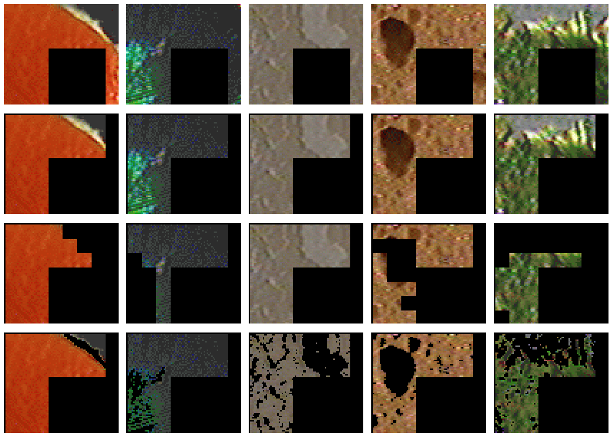
The optimal AR coefficients can be estimated as the solution to the following least square problem:

$$\boldsymbol{\alpha}_{C\times 1} = \arg\min_{\alpha} \|\mathbf{y}_{S\times 1} - \mathbf{X}_{S\times C}\boldsymbol{\alpha}_{C\times 1}\|^2 \quad (2)$$

where $\boldsymbol{\alpha}$ $\left(\boldsymbol{\alpha} \in \mathbb{R}^C\right)$ is a vector containing the AR coefficients (cf. Fig. 2). $\mathbf{y}$ $(\mathbf{y} \in \mathbb{R}^S)$ denotes the known samples I in the sub-training area (training area subtracted from left and top margin of size c_x and c_y) and $\mathbf{X}$ $\left(\mathbf{X} \in \mathbb{R}^{S\times C}\right)$ represents the neighbouring sample matrix for each of the samples in $\mathbf{y}$. C is the number of prediction coefficients and $S = s_x s_y$ the size of the sub-training area (the number of linear equations). Hence, eq. 2 can be solved with the closed-form solution:

$$\boldsymbol{\alpha} = (\mathbf{X}^T\boldsymbol{\alpha})^{-1}(\mathbf{X}^T\mathbf{y}). \quad (3)$$

As the set of coefficients α minimizes the model error in a least-square sense, samples that are unsuitable for completion in the current training area are assigned smaller coefficients,



(a) (b) (c) (d) (e)

Fig. 4. Pruning of the training area for (a) *orange peel*, (b) *peacock feather*, (c) *lettuce leaf*, (d) *sponge* and (e) *moss*. Examples of (top to down) input with $\Omega = 40 \times 40$; the non-stationary training area; the training area after applying the new block-based clustering criterion; and the training area after applying k-means clustering.

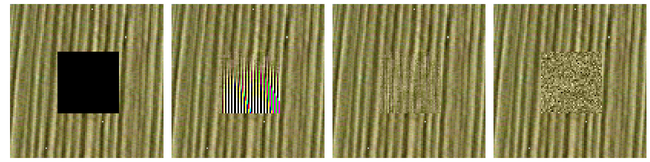
i.e. the AR model adapts to local texture characteristics. In case eq. 2 cannot be solved, due to non-invertible matrices $\mathbf{X}^T\mathbf{X}$, a pseudo inverse can be determined [20]. Once the AR coefficients are estimated, the standard deviation σ^2 of the innovation term $\epsilon(x, y)$ is calculated using the completion error normalized by the size of the sub-training area [8]:

$$\sigma^2 = \frac{||\mathbf{y}_{S\times 1} - \mathbf{X}_{S\times C}\boldsymbol{\alpha}_{C\times 1}||^2}{S}. \tag{4}$$

Depending on the completion scenario, the completed texture in Ω may still feature noticeable perceptual distortions, especially at the non-causal boundary borders. A post-processing based on Poisson cloning [19], focussing on these areas, is proposed to photometrically correct the seams. The following paragraphs present the other contributions (cf. dashed blocks in Fig. 1).

3.1. Optimization of the training area

The training area should contain valid information w.r.t the region Ω's unknown statistical properties. Fig. 3 (left diagram) shows an example of training area containing possible causal regions in a coding framework. According to the Markov Random Field theory, the texture must be stationary which is at this point not ensured over the training area. In this work, a new method to determine an optimized training area is proposed. Assuming that the prior training area is large enough (e.g. bigger than $3 \times \Omega$), we propose to remove unreliable parts in a fast and efficient manner. The causal area surrounding Ω is divided into blocks ($b_x \times b_y$) that are to be clustered into a stationary sub-set. The mean (μ) and variance (δ^2) of each block is determined (Gaussianity assumption) and clustering is operated based on the similarity of both features. For that, similarity thresholds t_μ and t_δ are introduced for μ- and δ^2-based comparisons respectively. As a result, a set of segments are obtained. The largest region is chosen as the validated training area (cf. Fig. 3 b). Fig. 4 depicts the final training area (third line) compared with k-means clustering [21].



(a) (b) (c) (d)

Fig. 5. Influence of the regularization with a model order $C = 15$ on *corn husk*. (a) input $\Omega = 40\times 40$. Results (without post-processing) with $S = 841$, $C = 15$ and (b) $\lambda = 0.1$, (c) $\lambda = 200$ and (d) $\lambda = 5e5$.

3.2. AR consistency criterion

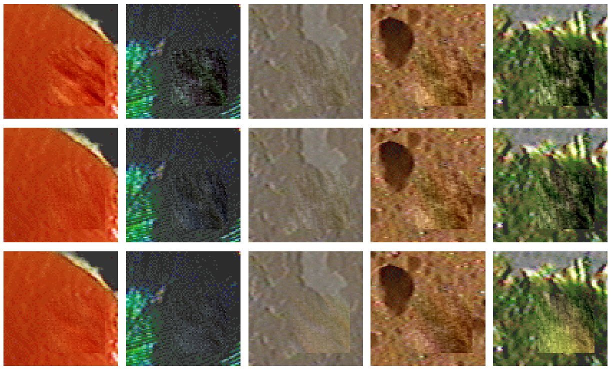
The best AR settings may still be a bad compromise, since texture surrounding Ω is finite. Moreover, the estimated AR coefficients may overfit the training data. Such inadequate AR parameters may lead to an erroneous propagation of the existing texture. In this work, an AR consistency assessment criterion is proposed to detect incorrect synthesis. It is assumed that the properties of the final result and those of the initialization area should be similar. If I_{min} and I_{max} are respectively the lowest and the highest initialization sample values, the completion is considered as unsuccessful if

$$\begin{cases} \hat{I}(x,y) & < & I_{min} - \tau \\ & or & \\ \hat{I}(x,y) & > & I_{max} + \tau \end{cases} \quad \text{with } (x,y) \in \Omega, \tag{5}$$

where τ is a threshold value, that allows a small deviation from I_{min} and I_{max}. This is a quite simple criterion that is motivated by the observation that AR distortions typically lead to gross chromatic variations that extremely deviate from the spatial context. In case of erroneous completion result, it is advised to use a "regularization procedure" [22] to make the system yield a different set of coefficients.

Regularization involves introducing additional parameter(s) in order to solve an ill-posed problem. By minimizing the augmented error function instead of the error on the image data, complex models can be penalized. In detail, a new parameter λ is defined that allows us to regularize the coefficients α, so that the variance of α is decreased, preventing overfitting. The new least square problem in eq. 2 can be expressed as:

$$\boldsymbol{\alpha}_{C\times 1} = \arg\min_{\alpha} \left[\|\mathbf{y}_{S\times 1} - \mathbf{X}_{S\times C}\boldsymbol{\alpha}_{C\times 1}\|^2 + \lambda \|\boldsymbol{\alpha}_{C\times 1}\|^2 \right]. \tag{6}$$



(a) (b) (c) (d) (e)

Fig. 6. Stationarity of the training area. Results (top to down); the non-stationary training area; the training area after applying the new block-based clustering criterion; and the training area after applying k-means clustering.

Methods	**Average time (s)**	**Loss factor**
AR	**0.199**	**1.0**
AR with post-processing	0.303	1.5
Priority-based [10]	38.642	194.2
FSE [23]	3.517	17.7

Table 1. Average run-time performances of different frameworks over the whole dataset.

Hence, eq. 6 can be estimated with the closed-form solution:

$$\boldsymbol{\alpha} = (\mathbf{X}^T\boldsymbol{X} + \lambda\boldsymbol{U})^{-1}(\mathbf{X}^T\mathbf{y}). \tag{7}$$

where $\boldsymbol{U} \in \mathbb{R}^{C\times C}$ represents the unit matrix. Fig. 5 depicts the results obtained by applying the regularization criterion. Therefore, the completion can be improved by varying the regularization parameter λ, which has to be optimized to avoid overfitting (cf. Fig. 5 b) and underfitting (cf. Fig. 5 d).

The next section presents subjective results of these contributions and a comparison with state-of-the-art algorithms.

4. RESULTS

For experiments, the data set has been chosen from the Columbia Utrecht Reflectance and Texture Database (CUReT) [24] to cover a broad spectrum of texture characteristics. It contains the 20 images: *rough plastic, plaster, rough paper, artificial grass, cork, sponge, lettuce leaf, loofa, limestone, ribbed paper, straw, corduroy, stones, corn husk, white bread, soleirolia, orange peel, peacock feather, tree bark* and *moss*.

All tests were performed using the constellation shown in Fig. 3 with $\Omega = 40 \times 40$, a sub-training size of $S \approx 800$ before block discarding. $C = 15$ and $\tau = 30$ were experimentally chosen, providing a good compromise between quality and computational complexity. $\lambda \approx 100$ gives good results for our test set (Fig. 5). Fig. 6 shows the quality improvement resulting from new stationary training areas. The results are computed from the training areas depicted in Fig. 4.

(a)
(b)
(c)
(d)

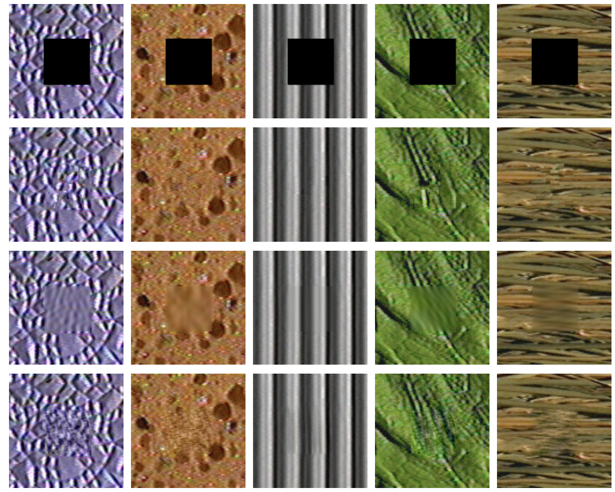


Fig. 7. Subjective results for (from left to right) *rough plastic, sponge, ribbed paper, lettuce leaf, straw*. (a) Input with $\Omega = 40 \times 40$. Completed results with the (b) priority-based [10], (c) FSE [23] algorithms. (d) Results of the AR texture completion with $C = 15$ and $S = 841$ and post-processing.

The proposed framework is subjectively compared in Fig. 7 to the priority-based [10] and the Frequency Selective Extrapolation (FSE) [23] methods. The first is executed with a patch size of 9×9 samples, as recommended in [10]. For the parametrical FSE approach, the settings proposed in [23] were used. The results achieved by the FSE are significantly blurrier as the proposed AR framework, but the priority-based method [10] often provides better visual results.

However, the AR method requires low computational effort. In particular, it is approximately 194 and 18 times faster than the priority-based [10] and FSE [23] approaches. The averaged run times in Table 4 are estimated over all test images. Gains are calculated in relation to the performance time of the AR approach without post-processing. Using the post-processing module adds only a small complexity overhead, compared to the competing completion methods.

5. CONCLUSION

This paper proposed an improved 2D AR framework for texture completion. The training area is optimized to contain a stationary texture only. In case of detected errors, an AR consistency criterion enables the scheme to regularize the set of parameters. Finally, Poisson cloning is used to remove the remaining artefacts. Presented results legitimate the use of these new tools for texture completion in case a fast approach is required. In future work, we will address the extension of this completion approach to the 2D+t domain.

6. REFERENCES

[1] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, Greg Turk, et al., "State of the art in example-based texture synthesis," in *Eurographics 2009, State of the Art Report, EG-STAR*, 2009, pp. 93–117.

[2] M Bertalmío, V Caselles, S Masnou, and G Sapiro, "Inpainting," *Encyclopedia of Computer Vision, Springer*, 2011.

[3] Javier Portilla and Eero P Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–70, 2000.

[4] David J Heeger and James R Bergen, "Pyramid-based texture analysis/synthesis," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 229–238.

[5] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[6] R Chellappa and RL Kashyap, "Texture synthesis using 2-d noncausal autoregressive models," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 33, no. 1, pp. 194–203, 1985.

[7] Jitendra K Tugnait, "Estimation of linear parametric models of nongaussian discrete random fields with application to texture synthesis," *Image Processing, IEEE Transactions on*, vol. 3, no. 2, pp. 109–127, 1994.

[8] Martin Szummer and Rosalind W Picard, "Temporal texture modeling," in *Image Processing, 1996. Proceedings., International Conference on*. IEEE, 1996, vol. 3, pp. 823–826.

[9] Li-Yi Wei and Marc Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 479–488.

[10] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama, "Region filling and object removal by exemplar-based image inpainting," *Image Processing, IEEE Transactions on*, vol. 13, no. 9, pp. 1200–1212, 2004.

[11] Michael Ashikhmin, "Synthesizing natural textures," in *Proceedings of the 2001 symposium on Interactive 3D graphics*. ACM, 2001, pp. 217–226.

[12] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick, "Graphcut textures: image and video synthesis using graph cuts," in *ACM Transactions on Graphics (TOG)*. ACM, 2003, vol. 22, pp. 277–286.

[13] Patrick Ndjiki-Nya, Martin Köppel, Dimitar Doshkov, and Thomas Wiegand, "Automatic structure-aware inpainting for complex image content," in *Advances in Visual Computing*, pp. 1144–1156. Springer, 2008.

[14] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[15] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *Image Processing, IEEE Transactions on*, vol. 10, no. 8, pp. 1200–1211, 2001.

[16] Anat Levin, Assaf Zomet, and Yair Weiss, "Learning how to inpaint from global image statistics," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 305–312.

[17] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher, "Simultaneous structure and texture image inpainting," *Image Processing, IEEE Transactions on*, vol. 12, no. 8, pp. 882–889, 2003.

[18] Koichiro Deguchi, "Two-dimensional auto-regressive model for analysis and sythesis of gray-level textures," *Proc. of the 1st Int. Sym. for Science on Form, General Ed. S. Ishizaka, Eds. Y. Kato, R. Takaki, and J. Toriwaki*, pp. 441–449, 1986.

[19] Patrick Pérez, Michel Gangnet, and Andrew Blake, "Poisson image editing," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 313–318, 2003.

[20] Gene Golub and William Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *Journal of the Society for Industrial & Applied Mathematics, Series B: Numerical Analysis*, vol. 2, no. 2, pp. 205–224, 1965.

[21] Christopher M Bishop, *Neural networks for pattern recognition*, Oxford university press, 1995.

[22] Ethem Alpaydin, *Introduction to machine learning*, MIT press, 2004.

[23] André Kaup, Katrin Meisinger, and Til Aach, "Frequency selective signal extrapolation with applications to error concealment in image communication," *AEU-International Journal of Electronics and Communications*, vol. 59, no. 3, pp. 147–156, 2005.

[24] Kristin J Dana, Bram Van Ginneken, Shree K Nayar, and Jan J Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics (TOG)*, vol. 18, no. 1, pp. 1–34, 1999.