WAVELET CONTRAST-BASED IMAGE INPAINTING WITH SPARSITY-DRIVEN INITIALIZATION

Philipp Tiefenbacher *Michael Sirch* *Mohammadreza Babaee* *Gerhard Rigoll*

Institute for Human-Machine Communication, Technical University of Munich

ABSTRACT

Image inpainting is the task of removing undesired objects or flaws in images. This work advances an exemplar-based global optimization image inpainting algorithm. For that purpose, the inpainting area is iteratively refined through the minimization of a cost function. The minimization outcome depends on the initial values of the inpainting area. We compare three initialization methods with a new sparsity-driven approach. Lastly, we propose the new wavelet contrast costs which increase the inpainting quality. Wavelet contrasts reduce computational complexity in comparison to wavelet histograms while preserving their ability of measuring the density of image texture.

Index Terms— Wavelet transforms, inpainting, initialization, algorithm

1. INTRODUCTION

In an image inpainting task, a part of an input image, usually called target region, is filled. The target region's content can either be unknown or undesired and should be replaced. The perceptibility of the inpainting result depends largely on the image content as well as on target region size and shape. There are algorithms that are able to inconspicuously synthesize small target regions within single-texture images. When the target region includes different textures, the inpainting tasks becomes more complex. Tiefenbacher et al. [1] measured inpainting quality of different algorithms. Their findings exhibit that the algorithms usually fail to unrecognizably inpaint large target regions, e.g., 10% of the source image. Consequently, there is still high potential for improving inpainting algorithms.

They also showed that exemplar-based global optimization algorithms, in particular [2], produce the best results on their test bench. Therefore, the baseline for our improvements and comparisons in inpainting quality is the algorithm of Herling and Broll [2]. Our findings, however, can be applied to any exemplar-based inpainting algorithm.

The contributions of this work are twofold: Four initialization methods for the global optimization problem are evaluated, and a new cost metric measuring texture is introduced and compared.

2. RELATED WORK

Our work is directly based on exemplar-based inpainting, which aims at restoring missing texture [3]. A thorough overview of exemplar-based inpainting can be found in [4]. In practice, a square patch $\Psi_{\boldsymbol{p}} \subset \Omega$ around pixel $\boldsymbol{p}$ is defined

$$\Psi_{\boldsymbol{p}} = \{\boldsymbol{p}_i \mid \boldsymbol{p}_i = \boldsymbol{p} + \boldsymbol{k},\ \boldsymbol{k} \in \mathcal{N}\}\,, \tag{1}$$

with $\mathcal{N} \subset \mathbb{Z}^2$ being the relative coordinates of the neighborhood. Then, the intensity values of all pixels in $\Psi_{\boldsymbol{p}}$ are compared to patches of the same size in the source image.

Greedy algorithms [3, 5] process each pixel in the target region $\mathcal{T}$ only once. In each step, they select the currently best possible match from the source region $\mathcal{S}$. The final result for the overall target region $\mathcal{T}$, however, may not be the best solution, since these approaches consider only neighboring target pixels which have already been filled.

Therefore, global optimization methods [2, 6, 7], to which our algorithm belongs, formalize the inpainting task as mathematical minimization problem. This minimization problem is solved iteratively by processing each pixel in the target region $\mathcal{T}$ multiple times. The optimization parameter is the mapping of patches in the source region $\mathcal{S}$ to pixels in the target region $\mathcal{T}$. It is stored in the correspondence function $\varphi : \mathcal{T} \to \mathcal{S}$.

The cost term $c(\boldsymbol{p}, \varphi)$ of exemplar-based algorithms is generally a combination of weighted sums. The cost function implemented by Herling and Broll [2] and used in our algorithm consists of terms which compare appearance and spatial distance

$$c(\boldsymbol{p}, \varphi) = \alpha \cdot c_{app.}(\boldsymbol{p}, \varphi) + (1 - \alpha) \cdot c_{spa.}(\boldsymbol{p}, \varphi). \tag{2}$$

The appearance cost is calculated via

$$c_{app.}(\boldsymbol{p}, \varphi) = \sum_{v \in N_a} d_a[u(\boldsymbol{p}+v), u(\varphi(\boldsymbol{p})+v)] \cdot w_a(\boldsymbol{p}+v) \tag{3}$$

with N_a defining the relative coordinates to the neighborhood. We set the neighborhood N_a to 5x5 patches around the original patch $\Psi_{\boldsymbol{p}}$. v is the offset vector and $w_s(v)$ describes the weight map for the neighboring pixels which sums up to one. The distance measure $d_a(\cdot, \cdot)$ computes the sum of squared differences (SSD) between the pixel values $u(\boldsymbol{p})$. The formulation of the spatial cost is

$$c_{spat.}(\boldsymbol{p}, \varphi) = \sum_{v \in N_s} d_s[\varphi(\boldsymbol{p}) + v, \varphi(\boldsymbol{p} + v)] \cdot w_s(v). \tag{4}$$

N_s holds the neighborhood pixels, which is the set of 3x3 pixels around the original pixel $\boldsymbol{p}$. w_s is another weight map. The distance function $d_s(\cdot,\cdot)$ measures the euclidean distance between two pixels taking the upper distance clamp τ_s into account. We set this distance clamp τ_s to 200, which was recommended by Herling and Broll [8].

Then, the solution of the inpainting is calculated iteratively due to the mutual dependencies between the target pixels. First, an initial mapping φ^0 assigns color values to each target pixel. This processing step influences the inpainting result and is evaluated in Section 3.

After initialization, the algorithms try to improve the correspondence function φ^i in each iteration i by solving

$$\arg\min\ c(\boldsymbol{p}_i, \varphi^i), \quad \forall \boldsymbol{p}_i \in \mathcal{T}. \tag{5}$$

The intermediate correspondences of the last iteration constitute the cost term values of other target pixels. Ideally, φ^i converges to the ideal mapping φ after multiple iterations.

3. INPAINTING INITIALIZATION

Initialization of the target region is an important step for exemplar-based inpainting algorithms. A poor initialization terminates more frequently in a local minimum, which may deviate largely from an optimal solution. Furthermore, the initialized, low-resolution target region already contains a structure. The patches from the source region are compared to the content of the target region. Thus, it is likely that the initial structure of the target region persists until the final inpainting result, making the initialization a crucial step.

Different approaches for initialization exist but have not been compared yet. Barnes et al. [9] chose target region values randomly, Herling and Broll [10] interpolated border values with a 3x3 morphological erosion filter of the target region to the inside. Other algorithms [6, 7] incorporated greedy inpainting using an onion-peel scheme [11, 12] for the initialization of the target region. Some inpainting algorithms [13, 14] applied the onion-peel scheme also for the final inpainting.

We present and compare four different methods for the initialization of inpainting: random [9], morphological erosion [10] and two greedy-based ones. Since the concepts for random and erosion initialization are simple, we only explain the greedy initializations in more detail.

3.1. Greedy Initialization

The initialization step employs the same cost terms as in the global optimization steps afterwards. This ensures consistency within the iteration steps.

Onion-Peel Initialization The onion-peel scheme defines the inpainting order of the target region pixels in an efficient manner. First, a random pixel is selected at the border between known and unknown pixels. Then, the filling order of the target region is layer-wise from the outside to the center such as the layers of an onion.

Sparsity-Driven Initialization The sparsity-driven approach is derived from Xu and Sun [5], whose filling order prioritized sparsity. The new greedy initialization order completes edges and corners first while moving from the boundaries inward.

Target patches are compared to their neighborhood. The similarity between the patches $\Psi_{\boldsymbol{p}}$ and $\Psi_{\boldsymbol{q}}$ is defined as

$$\omega_{\boldsymbol{p},\boldsymbol{q}} = \frac{1}{Z(\boldsymbol{p})} \exp\left(-\frac{d(\Psi_{\boldsymbol{p}}, \Psi_{\boldsymbol{q}})}{\sigma^2}\right), \tag{6}$$

where $d(\cdot,\cdot)$ is the mean-squared distance between two patches (involving only filled pixels). $Z(\boldsymbol{p})$ is a normalization factor that $\sum_{\boldsymbol{q}\in N_f(\boldsymbol{p})} \omega_{\boldsymbol{p},\boldsymbol{q}} = 1$ and σ is a constant, which is set to 5.0. The sparseness $S(\boldsymbol{p})$ can now be calculated from the distribution of the similarity coefficients $\omega_{\boldsymbol{p},\boldsymbol{q}}$

$$S(\boldsymbol{p}) = \sqrt{\frac{|N_f(\boldsymbol{p})|}{|N(\boldsymbol{p})|} \sum_{\boldsymbol{q}\in N_f(\boldsymbol{p})} \omega^2_{\boldsymbol{p},\boldsymbol{q}}}. \tag{7}$$

The neighborhood $N(\boldsymbol{p})$ is a quadratic window centered at the pixel $\boldsymbol{p}$, whereas $N_f(\boldsymbol{p})$ is defined as the neighborhood pixels which represent completely filled patches (including the source region). If the similarity is low, the target pixel is at an inhomogeneous position, i.e., it is at the border of a unique structure and should be synthesized with a higher priority.

3.2. Results of Initialization

We replaced the erosion-based initialization of the algorithm by Herling and Broll [10] with the random, greedy onion-peel, and greedy sparsity-driven initialization methods. Then, inpainting is performed on the database images of [1] using each of the four initialization methods.

Figure 1 shows a qualitative comparison for inpainting a compact region [1]. The initialization on the coarsest layer is depicted for each of the three methods on the top. The random initialization consists of six pyramid layers, which fulfills the three-pixels-to-the-border recommendation of Herling and Broll [8]. The number of layers can be set to five for the two greedy-based methods, since a higher pyramid does not improve the result further.

Both random and onion-peel initializations are detectable, since the street texture is inserted instead of the house or the tree texture. The morphological erosion in Figure 1a) achieves a quite satisfying result, since mostly tree textures are inserted and only a very small part of the street. However, the synthesized house frontage does not fill in well. The sparsity-driven initialization shows the best result.

4. WAVELET CONTRAST COSTS

Herling and Broll [2] correlated only the color information of the patches. This often results in erroneously synthesized textures [7] since the texture frequency is not taken into account.

We want to consider the image structure via wavelets in the cost metric while preserving a fast inpainting algorithm.

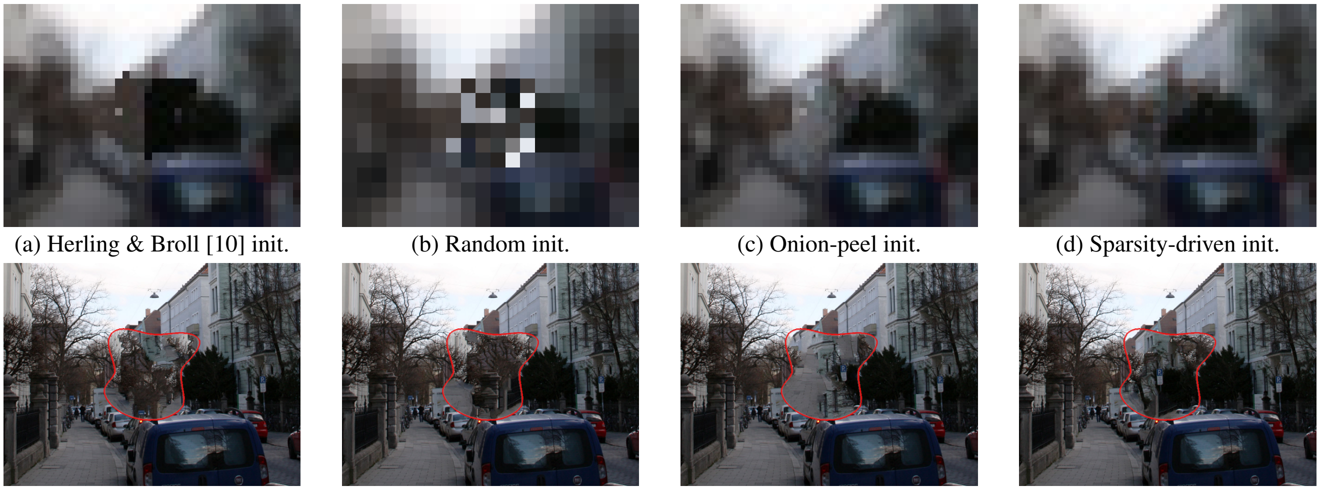

(a) Herling & Broll [10] init.
(b) Random init.
(c) Onion-peel init.
(d) Sparsity-driven init.

Fig. 1: Different initialization methods applied to a complex test image with a compact mask. The inpainting results with target region indication (red) are shown at the bottom.

Heeger and Bergen [15] showed that wavelet histograms are able to analyze texture for texture synthesis tasks. Our contribution is to reformulate the wavelet histograms as a cost function by considering the findings of Wang and Simoncelli [16] to accelerate the wavelet calculation.

4.1. Wavelet Histograms

Generally, wavelet histograms decompose the image through a steerable pyramid decomposition [17]. Each layer of such a pyramid contains a specific frequency band of the image luminance. The layers are divided into multiple subbands. Each subband captures frequencies in another direction. After the creation of the subband images, the distributions of the wavelet coefficients for each subband are retrieved via the construction of histograms. Finally, the absolute value of the Kullback-Leibler divergence (KLD) [18] quantifies the difference between the histogram distributions.

We propose two modifications to the steerable pyramid decomposition to adapt it for inpainting: 1) we use only four directions and the second pyramid layer. The second pyramid layer features the highest frequency. The lower frequency subbands contain no information about fine texture details and are ignored. 2) Histogram computation and comparison is performed patch-wise (33x33), which is essential for integrating the wavelet histograms in the cost function of the exemplar-based inpainting algorithm.

We extracted three different textures from Figure 3 and computed the wavelet coefficients of the first subband, which measure the frequency in x-direction (s_0). Figure 2 illustrates the textures as well as the wavelet coefficients. A variation between the different subband images is visible. For instance, the subband image of the gravel texture 2c) consists of many brighter and darker spots in the image, which implies higher frequencies. The metal subband 2a) shows less contrast, since the metal texture is rather smooth.

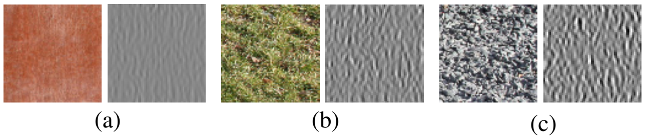

(a)
(b)
(c)

Fig. 2: A metal (a), grass (b) and gravel (c) texture with the result of the first wavelet subband.

This approach, however, suffers from the processing time of up to several minutes due to the expensive histogram creation and KLD. Wang and Simoncelli [16] took advantage of the fact that distributions of wavelet coefficients can be approximated by a generalized Gaussian distribution defined by only two parameters [19, 20]. Their comparison is still based on the probability distribution with KLD.

4.2. Wavelet Contrasts

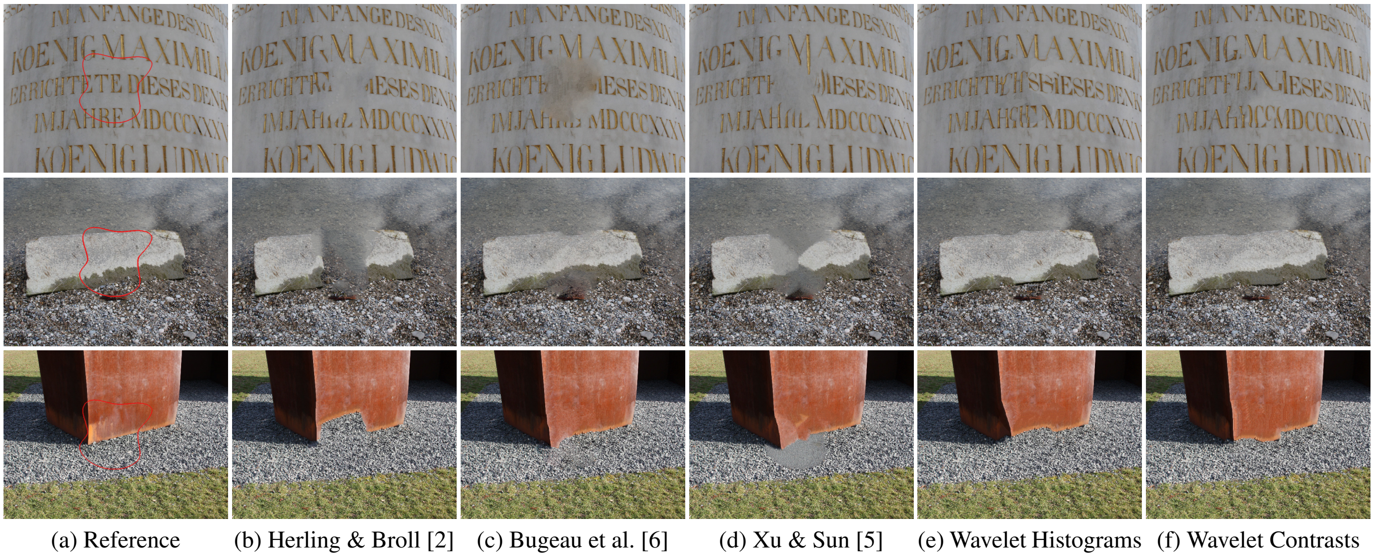
We improve this by taking only the standard deviation of wavelet coefficients into account and call this the wavelet contrasts. The wavelet contrasts are computed for every subband image. The SSD between the wavelet contrasts of different patches extends the existing cost function. No histogram creation is required anymore.

In detail, a wavelet contrast pyramid is constructed. For that reason, the four highest-frequency oriented subbands s_0, s_1, s_2, and s_3 are retrieved from the full-size input image in the first part of a steerable pyramid decomposition. We choose the subband filters of [21]. Then, the wavelet contrast $\sigma_w^i(\boldsymbol{p})$ is defined as the root mean square contrast [22] of the neighborhood $\nu_{\boldsymbol{p}}$ of each source pixel $\boldsymbol{p}$ for each subband s_i:

$$\sigma_w^i(\boldsymbol{p}) = \sqrt{\frac{1}{|\nu_{\boldsymbol{p}}|-1}\sum_{\boldsymbol{q}\in\nu_{\boldsymbol{p}}}\left[s_i(\boldsymbol{q})-\overline{s}_i(\boldsymbol{p})\right]^2}. \quad (8)$$

The subband means $\overline{s}_i(\boldsymbol{p})$ are derived with

$$\overline{s}_i(\boldsymbol{p}) = \frac{1}{|\nu_{\boldsymbol{p}}|}\sum_{\boldsymbol{q}\in\nu_{\boldsymbol{p}}} s_i(\boldsymbol{q}). \quad (9)$$



(a) Reference (b) Herling & Broll [2] (c) Bugeau et al. [6] (d) Xu & Sun [5] (e) Wavelet Histograms (f) Wavelet Contrasts

Fig. 3: Image (a) is the original image with marked target region. Images (b)-(f) show the results of different algorithms.

Each source pixel $\boldsymbol{p}$ receives a four-dimensional wavelet contrast vector $\boldsymbol{\sigma}_w$ consisting of the wavelet contrast values of each subband at this location:

$$\boldsymbol{\sigma}_w(\boldsymbol{p}) = \left[\sigma_w^0(\boldsymbol{p}), \sigma_w^1(\boldsymbol{p}), \sigma_w^2(\boldsymbol{p}), \sigma_w^3(\boldsymbol{p})\right] . \tag{10}$$

A four-channel image includes the contrast vector of each pixel. We subsample this image to create a wavelet contrast pyramid. During the inpainting iterations, the wavelet contrast vectors are inpainted concurrently to the color information. The new wavelet cost term $c_w(\boldsymbol{p}, \varphi)$ is calculated with

$$c_w(\boldsymbol{p}, \varphi) = \frac{1}{|\mathcal{N}|} \sum_{\boldsymbol{k} \in \mathcal{N}} \left\| \boldsymbol{\sigma}_{wav.}(\boldsymbol{p} + \boldsymbol{k}) - \boldsymbol{\sigma}_{wav.}(\varphi(\boldsymbol{p}) + \boldsymbol{k}) \right\|^2 . \tag{11}$$

4.3. Results of Wavelet Contrast-based Inpainting

The wavelet contrast costs are added to the inpainting algorithm of Herling and Broll [2], enhancing the appearance and spatial cost terms of Equation (2). The weights α_i for the costs are set to 0.9, 0.1 and 1.8 for the appearance, spatial and wavelet cost terms, respectively. The inpainting initilzation is done via the presented sparsity-driven approach. We use the inpainting database of [1] for all comparisons.

Figure 3 compares our inpainting results to the algorithms of Bugeau et al. [6] and Xu and Sun [5]. Bugeau et al. [6] combines PDE-based [23] with exemplar-based inpainting. The algorithm by Xu and Sun [5] applies greedy inpainting [3, 24] and creates the missing parts by combining several patches, which is a common approach [25].

Figure 3e) shows the result with wavelet histograms instead of wavelet contrasts. The wavelet histograms consist of 50 bins and are weighted with 36. Our new results incorporate the sparsity-driven initialization and the normalization step during pyramid creation. The wavelet methods in Figure 3e) and f) reconstruct the inpainting regions of the images in the first and second row of Figure 3 correctly, whereas the other algorithms fail except Bugeau et al. [6] in the second row. However, the edge of the stone is sharper with our algorithm.

The third row of Figure 3 shows that the gravel texture has been inserted into the brown metal sculpture for image 3b). The wavelet contrast term prevents this cloning. Bugeau et al. [6] leads to a quite appealing result even though the frequency of the gravel texture is too high. The inpainted texture by Xu and Sun [5] is too smooth and therefore noticeable.[1]

Wavelet contrast achieves similar inpainting quality as wavelet histograms while reducing the computational time drastically. The wavelet contrast term needs only 2-5 s additional time for images with 640x480 pixels and a target region covering 10% of the image. The wavelet histograms delay the inpainting task by several minutes.

5. CONCLUSION AND OUTLOOK

We advanced a state-of-the-art exemplar-based inpainting algorithm. First, we showed that the initialization methods derived from texture synthesis are superior to a random initialization. Initialization schemes which consider image structure, such as the onion-peel and sparsity-driven schemes, guide the inpainting process and lead to an improved result. The sparsity-driven and onion peel schemes perform equally. We recommend to apply at least the onion-peel scheme, since it introduces lower computational complexity than the sparsity-driven scheme.

Second, the newly introduced wavelet contrasts can improve the inpainting results. Both wavelet costs allow for considering structural density, which complements the appearance and sparsity terms of the cost function. The wavelet contrasts reduce the computational effort in comparison to wavelet histograms while maintaining the improved visual quality.

[1] This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The material comprises a video which illustrates further inpainting results and has a size of 12.3 MB.

6. REFERENCES

[1] P. Tiefenbacher, V. Bogischef, D. Merget, and G. Rigoll, "Subjective and objective evaluation of image inpainting quality," in *Proceedings of ICIP*. 2015, pp. 447–451, IEEE.

[2] J. Herling and W. Broll, "PixMix: A real-time approach to high-quality diminished reality," in *Proceedings of ISMAR*. IEEE, 2012, pp. 141–150.

[3] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *Proceedings of CVPR*. IEEE, 2003, pp. 721–728.

[4] P. Buyssens, M. Daisy, D. Tschumperle, and O. Lézoray, "Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1809–1824, 2015.

[5] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1153–1165, 2010.

[6] A. Bugeau, M. Bertalmío, V. Caselles, and G. Sapiro, "A comprehensive framework for image inpainting," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2634–2645, 2010.

[7] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video inpainting of complex scenes," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 1993–2019, 2014.

[8] J. Herling and W. Broll, "High-quality real-time video inpainting with PixMix," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 6, pp. 866–879, 2014.

[9] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "PatchMatch: a randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 24, 2009.

[10] J. Herling and W. Broll, "Advanced self-contained object removal for realizing real-time diminished reality in unconstrained environments," in *Proceedings of ISMAR*. IEEE, 2010, pp. 207–212.

[11] R. Bornard, E. Lecan, L. Laborelli, and J.-H. Chenot, "Missing data correction in still images and image sequences," in *Proceedings of MULTIMEDIA*. ACM, 2002, pp. 355–361.

[12] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 303–312, 2003.

[13] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of ICCV*. IEEE, 1999, pp. 1033–1038.

[14] Y. Zhang, J. Xiao, and M. Shah, "Motion layer based object removal in videos," in *Proceedings of WACV/MOTIONS*. IEEE, 2005, pp. 516–521.

[15] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proceedings of SIGGRAPH*. ACM, 1995, pp. 229–238.

[16] Z. Wang and E. P. Simoncelli, "Reduced-reference image quality assessment using a wavelet-domain natural image statistic model," *SPIE Human Vision and Electronic Imaging X*, vol. 5666, no. 149, pp. 149–159, 2005.

[17] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 587–607, 1992.

[18] S. Kullback and R. A. Leibler, "On information and sufficiency," *JSTOR The annals of mathematical statistics*, vol. 22, pp. 79–86, 1951.

[19] P. Moulin and J. Liu, "Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors," *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 909–919, 1999.

[20] E. P. Simoncelli and E. H. Adelson, "Noise removal via Bayesian wavelet coring," in *Proceedings of ICIP*. IEEE, 1996, pp. 379–382.

[21] A. Karasaridis and E. Simoncelli, "A filter design technique for steerable pyramid image transforms," in *Proceedings of ICASSP*. IEEE, 1996, pp. 2387–2390.

[22] E. Peli, "Contrast in complex images," *OSA Journal of the Optical Society of America A*, vol. 7, no. 10, pp. 2032–2040, 1990.

[23] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of SIGGRAPH*. ACM, 2000, pp. 417–424.

[24] P. Perez, M. Gangnet, and A. Blake, "Patchworks: Example-based region tiling for image editing," Tech. Rep., Microsoft Res., 2004.

[25] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007.