Symmetry-Based Façade Repair

Przemyslaw Musialski* Peter Wonka‡ Meinrad Recheis* Stefan Maierhofer*
Werner Purgathofer*

*VRVis Research Center, Vienna, Austria
{pm,recheis,sm,wp}@vrvis.at

‡Arizona State University, Tempe, USA
pwonka@gmail.com

Abstract

In this paper we address the problem of removing unwanted image content in a single orthographic façade image. We exploit the regular structure present in building façades and introduce a diffusion process that is guided by the symmetry prevalent in the image. It removes larger unwanted image objects such as traffic lights, street signs, or cables as well as smaller noise, such as reflections in the windows. The output is intended as source for textures in urban reconstruction projects.

Figure 1: The input image on the left contains a traffic light and several cables. To the right we show the result of our algorithm with the unwanted objects successfully removed.

1 Introduction

This paper introduces a special image-processing method based on *symmetry propagation*. The proposed algorithm takes a single ortho-rectified façade image as input and tries to remove unwanted content, such as wall impurities, cables, and street signs (Figure 1). While this approach is similar to image in-painting in some respects, our goal and methodology are fairly different. First, we do not want to manually mark the irregularities by hand before removing them, but we would like to identify them automatically. Second, the focus of our algorithm is not a smooth transition or texture propagation from nearby regions, but structure propagation from detected symmetries. In principle, our algorithm is general and removes irregularities over a regular structure. While there are several potential applications for such an approach, the main motivation and probably most important use is the processing of façade images. Façade images are a vital component of three-dimensional urban reconstruction and we see applications in areas such as Internet and car-based mapping technology, urban simulation, and computer games.

Our contribution is twofold. First, we detect regular structures using a combination of Monte Carlo sampling and user interaction. Second, we use a diffusion-like process that tries to smooth across symmetries. It is a novel image processing technique that utilizes spatial symmetry in order to minimize asymmetric variations inherit in the image.

1.1 Related Work

In this section we review façade reconstruction approaches, symmetry detection, image repair, and in-painting methods. We limit our review to techniques that act on a single image.

Alegre and Dellaert [1] introduce a hierarchical context free grammar to obtain semantic description of buildings which is obtained by MCMC. Han and Zhu [9] detect regular rectangular structures in photographs of arbitrary scenes. Their approach combines bottom-up and top-down image interpretation by using an attribute grammar. Müller *et al.* [21] propose a hierarchical segmentation of façades inspired by shape grammars. Korah and Rasmussen [12] introduce a MCMC based technique to detect regular grids of windows in a MRF network. Further, they propose [13] a method to automatically complete façade images based on PCA. Apart from these highly specialized solutions there are more

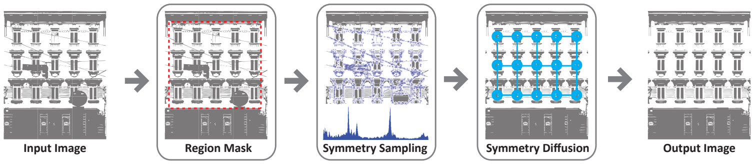

Figure 2: This figure shows an overview of the system. We take a single façade image as input I. For the case that the image does contain strongly asymmetric parts, we allow the user to define a region of interest. In this region we detect the dominant translational and reflective symmetries and propagate the symmetry over the image automatically.

general methods for the detection of similarity and symmetry in images. Bailey [3] shows that it is possible to detect repetitive image patterns by self-filtering in the frequency domain and Reisfeld *et al.* [26] introduce continuous symmetry transform for images. Liu *et al.* [15] detect crystallographic groups in repetitive image patterns using a dominant peak extraction method from the autocorrelation surface. Further succeeding approaches specialize on detecting affine symmetry groups in 2d [16, 18] or 3d [19, 25]. Recent follow-ups of these approaches introduce data-driven modeling methods like symmetrization [20] and 3d lattice fitting [23]. Further, recent image-processing approaches tend to utilize the detected symmetry of regular [11] and near-regular patterns [17, 14] in order to model new images. One further branch of image repair are in-painting algorithms. Example techniques include the use of PDEs [5, 22], fast matching distance maps [28] and MRFs [27]. In contrast to our approach these algorithms do not use symmetry to fill in the missing details.

1.2 Overview

The basic idea behind this paper is to exploit the repetitive occurrence of façade elements to reconstruct its clean and most plausible appearance. In order to handle the repetitive nature we introduce the concept of a global symmetry neighborhood, which provides some basic information about the actual image. While there is conceptually no limitation on the topological complexity of the neighborhood-graph, we currently assume that the supplied image has strong translational and reflective repetitive elements as in the case of a building.

In section 2 we introduce a basic method to automatically determine the dominant symmetry in an image. The method is based on a Monte Carlo importance sampling strategy of image patch pairs and histogram evaluation and it yields the translational and reflective-translational symmetry in the rectified image.

In section 3 we introduce a novel method that utilizes the inherent symmetry in order to reconstruct the façade image. Missing or occluded elements, clutter and damage as well as small perspective distortion are dislodged and replaced by the information that can be accessed over the symmetry neighborhood in the entire image.

In section 4 we present and discuss restored façade images and finally we conclude our work in section 5.

2 Symmetry Detection

In our context, we define symmetry as a transformation $\mathbf{T}$ on an image. Given a pixel location x of the input image I, we define $\mathbf{T}$ such that

$$I(x) = I(\mathbf{T}(x)) \quad (1)$$

where $I(x)$ denotes the intensity or color vector at x. As $\mathbf{T}$ we consider the 2-dimensional translations and reflections along the x-axis.

The goal of this stage is to determine the parameters of dominant transformations in the image automatically. The proposed algorithm is a histogram voting scheme and a similar approach has been proposed by, e.g., Mitra *et al.* [19]. The peaks of the histograms identify translational and reflective offsets in the image which occur most often and can be considered dominant.

Similarity Measure. Our method is based on comparison of image features from different spatial regions. We resort to area-registration methods

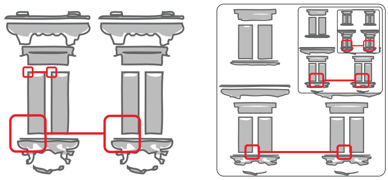

Figure 3: Left: depending on the window size, features can become ambiguous. Right: by comparing with the same patch size on different resolutions, the ambiguity can be resolved.

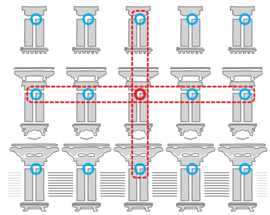

Figure 4: Symmetry Neighborhood. Blue circles denote symmetry neighbors of the red one.

based on the *local neighborhood* of a point of interest. The neighborhood of some image location i is usually denoted as a window around a pixel with the (maximum) radius r. It results in a square patch of size $w \times w$, where $w = 2r + 1$. It can also be interpreted as a vector $\mathbf{x}_i$ of dimension w^2.

In the literature, a common comparison operator is cross correlation of image patches, which is related to the dot-product between two given vectors $\mathbf{x}$ and $\mathbf{y}$: $\text{cc}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$. In order to compare positions with varying intensities, we compute the normalized cross correlation coefficient (NCC), where we subtract the mean of the intensities $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ of each patch $\mathbf{x}$ and $\mathbf{y}$ and normalize the vectors, respectively:

$$\text{ncc}(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \bar{\mathbf{x}})^T (\mathbf{y} - \bar{\mathbf{y}})}{\sqrt{(\mathbf{x} - \bar{\mathbf{x}})^2 (\mathbf{y} - \bar{\mathbf{y}})^2}} . \tag{2}$$

Multiresolution. Similarity between small windows such as 3×3 or 5×5 pixels can be computed very fast but the resulting measure is not very robust. When measuring patterns, the size of the pattern relative to the size of the measurement window is very important. If the pattern is too small or too large compared to the measurement window one will obtain ambiguous results (Fig. 3).

Rather than increasing the patch to improve robustness of the measure, a very efficient way is to combine the results of measurements on different scale levels of an image pyramid. This idea has been successfully used in many texture synthesis algorithms. It is computed by subsequently scaling the image with the factor s (in our case we use $s = \frac{1}{2}$ and cubic down-sampling). The similarity ς results from all scale levels which have been taken at the closest position to the original position in the unscaled picture and are then combined into the final result by taking the mean:

$$\varsigma(\mathbf{x}, \mathbf{y}) = \frac{1}{N_s} \sum_{k}^{N_s} \text{ncc}_k(\mathbf{x}, \mathbf{y}) \ , \tag{3}$$

where N_s is the number of scales and ncc_k operates on the k-th scale of the input image I. The window size is kept constant, as shown in Fig. 3 right hand side. In our empirical tests we determined, that a good trade-off between speed and robustness is a size of 15×15 pixels on 3 pyramid levels. This multiscale similarity measure is less error prone and very robust with respect to real-world noise while being relatively fast compared to using large similarity windows on the original image. Moreover it is practically independent of the size of the input images and the size of the patterns.

Monte Carlo Sampling. A naive deterministic approach to the problem of finding symmetry parameters would be to *exhaustively search* over all possible configurations. However, searching every possible combination is very computationally expensive, and time consuming for large images.

Using Monte Carlo sampling to obtain samples of the same data allows for a low-cost approximation of the expensive deterministic computation. Instead of comparing every pair of different locations, the Monte Carlo algorithm measures the similarity at a small number of random samples. We compute N similarity values $\varsigma(\mathbf{x}, \mathbf{x}_i)$ for all possible symmetry parameters, where i is determined randomly from the uniform distribution.

We improve the convergence and speed of the sampling by limiting the random samples to especially interesting points. In our implementation we use Harris corners [10]. This importance sampling restricts the algorithm to comparing only pixels which are known to contain local information.

Maximum Similarity Function. The maximum similarity function counts how many times a translation parameter δ is the best one such that its multiscale similarity is higher than that of any other offset at the sample location. A translation with a high number of hits represents a symmetry that covers a large image area and therefore is considered relevant. We define the maximum similarity histogram classification function for N random samples from a uniform distribution as:

$$h(\delta) = \sum_{i}^{N} \begin{cases} 1 & \text{if } \delta = \arg\max_{\delta} \varsigma(\mathbf{x}_i, \mathbf{x}_\delta) \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\delta \in D$. The range D is defined as a subset of all possible translations in the image.

The reflective symmetry detection can be accomplished using the same strategy, by mirroring the entire image and by respectively adjusting the translational offsets.

The symmetry parameters can be obtained by extracting the maxima from the histograms. In many cases the histograms include doubles, triples and higher multiples as well as other combinations of smaller offsets which are combined to one dominant offset.

3 Symmetry Propagation

Motivation. The symmetry propagation stage is the actual heart of our algorithm. Our idea behind this approach is motivated by the classical nonlinear diffusion filter as presented by Perona and Malik [24]. They presented a powerful method for discontinuity-preserving smoothing and denoising of images based on a divergence equation $\frac{\partial I}{\partial t} = \text{div}\left(g\left(\|\nabla I\|\right)\nabla I\right)$, where $g(x)$ is a flux-stopping function, which constrains the diffusion to pixels which have respectively small difference in range. It is usually of the form $g(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$ where x denotes the distance in the range. Later it has been shown [4] that this solution is also equivalent to the bilateral smoothing filter [29]. Hereby the basic idea is to apply a constrained Gaussian filtering to the image, such that steep range transitions become preserved. As a constraint an edge-penalty function has been introduced, which acts in principle the same way as the flux-stopping term mentioned above. The bilateral filter in a local neighborhood $\mathcal{N}_x$ of a pixel x in image I can be stated as:

$$I'_x = \frac{1}{W_x} \sum_{y \in \mathcal{N}_x} g_s\left(\|x-y\|\right) g_r\left(|I_x - I_y|\right) I_y \ , \tag{5}$$

where $W_x = \sum_y g_s(\|x-y\|) g_r\left(|I_x - I_y|\right)$ is a normalization term. There are two Gaussian functions in this equation: the usual g_s acting in the spatial domain and g_r applied on the range between the actually pixels values I_x and I_y. The subscripts s and r denote the standard deviations σ_s and σ_r of the respective Gaussians. The result obtained is an image smoothed only in regions where the range difference is small enough to be emphasized by G_r.

More recently, non-local means filtering has been proposed as a new class of solutions to the image denoising problem [6, 7, 8]. It is based on the observation that pixels with similar neighborhood usually appear quite often in an image. Non-local filtering exploits this observation by computing a noise-reduced image by weighted averaging of many similar pixels:

$$I'_x = \frac{1}{W_x} \sum_{y \in I} w(x, y)\, I_y \ . \tag{6}$$

The term $W_x = \sum_{y \in I} w(x, y)$ is the normalizing constant, such that all $w(x, y) \in [0, 1]$ and $\sum_y w(i, j) = 1$. The weights w are computed according to the squared Euclidean norm of local neighborhoods $\mathcal{N}_x$ and $\mathcal{N}_y$, respectively. Writing the respective neighborhoods as vectors $\mathbf{x}$ and $\mathbf{y}$, the distance penalty function once again has the form:

$$w(x, y) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{h^2}\right) .$$

The parameter h acts as a degree of filtering [7].

Iterative Symmetry Propagation. The approaches mentioned act (either locally or globally) on pixel neighborhoods in order to approximate a new image that is nearly noise-free, while a Gaussian penalty function is a common ingredient of these methods.

Our symmetry propagation algorithm is inspired by both bilateral- as well as non-local filtering. The

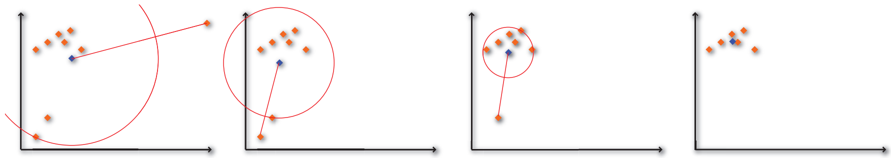

Figure 5: Recursive rejection of outliers. From left to right, at each step the point with the biggest distance to the mean (blue point) is removed and a new mean is computed, until the change is smaller than a given threshold.

main difference is that methods mentioned above aim at image repair by removing of noise that is a consequence of deficiencies in signal processing. We call it intrinsic noise. In contrast, our aim is the removal of both the intrinsic image noise as well as the extrinsic noise, e.g., traffic lights, cables, vegetation, missing elements and other interferences that are inherent in real world data. It is evident, that the second class of noise can be removed only under certain circumstances: (1) there must be enough repeated content in the image and (2) there must be a strategy how to localize that information. The first one is a general assumption that for each location to be repaired there is enough information in the image which can be reused. For the second we resort to the symmetry and expect that the feasible information is arranged in a manner which can be expressed in terms of symmetry transformations $\mathbf{T}$ of the form $I(x) = I(\mathbf{T}(x))$.

In section 2 we have presented an elementary symmetry detection scheme. Having determined the global symmetry, each pixel x in the image corresponds to a number of other pixels which can be addressed by the symmetry transformation $\mathbf{T}$ (see Equation 1). We shall refer to those pixels as the symmetry neighborhood $\mathcal{S}_x$ of the image location x. By the application of $\mathbf{T}$, it is possible to collect other neighbors and to obtain a set of points, which all correspond to a similarity in the image. Depending on the collection scheme, either all or only a subset of all possible symmetry neighbors can be accessed (see Fig. 4).

Having this information at a pixel, we now compute its actual consistency with its symmetry neighbors. Here we use different local neighborhood $\mathcal{N}_x$ as in the symmetry sampling stage from section 2. We have determined empirically, that sizes between 3×3 and 11×11 deliver reasonable results for our input images. With $\mathcal{N}_x$ as a vector $\mathbf{x}$ of intensity values we can compute the mean vector for all points $\mathbf{x}_i \in \mathcal{S}_x$ with $n = |\mathcal{S}_x|$ as $\bar{\mathbf{x}} = \frac{1}{n}\sum_i^n \mathbf{x}_i$.

If we would like to apply unconstrained symmetry propagation to the image, the actual new color value for the output pixel I'_x would be the middle element in $\bar{\mathbf{x}}$, which also equals the simple average color over all symmetry neighbors. But we are interested in some constraints, which will allow us to determine which of the pixels belong to the most symmetric façade image and which are potentially clutter. Following the first assumption that the majority of the pixels in $\mathcal{S}_x$ contain valid values, we introduce a scheme inspired by Expectation Maximization to reject outliers. It utilizes the fact that if there are more valid pixels they will also be more similar and thus lie denser to each other in the space defined by the local neighborhood vectors $\mathbf{x}_i$. In this case we define the outlier as the point which has the biggest distance to the mean $\bar{\mathbf{x}}$:

$$\mathbf{x}_i = \underset{\mathbf{x}_i \in \mathcal{S}_x}{\arg\max}\, K\left(\|\mathbf{x_i} - \bar{\mathbf{x}}\|^2\right) , \qquad (7)$$

where K is a Gaussian kernel. Now we can remove the i-th vector from $\mathcal{S}_x$ and recompute the mean $\bar{\mathbf{x}}$. We proceed iteratively until either the mean $\bar{\mathbf{x}}$ does not change more than a given threshold ϵ or only one point is left in $\mathcal{S}_x$. Using a sufficiently small ϵ this procedure delivers the most dense cluster of the symmetry neighborhood (see Fig. 5).

To determine the final color of the output pixel we additionally apply bilateral filtering over the local neighborhoods remained in $\mathcal{S}_x$ after the optimization, such that the output pixel value is:

$$I'_x = \frac{\sum_{\mathcal{S}} \sum_{\mathcal{N}} g_s(\|x-y\|) g_r\left(|I_x - I_y|\right) I_y}{W_x} , \qquad (8)$$

where W_x is an appropriate weight according to equation 5. We do this in order to smooth possibly remaining variations caused by inaccuracy of the symmetry transformation.

Image parts, which violate the detected symmetry are replaced by pixels which become amplified by strong symmetry. In case of strong asymmetries some of them can still remain in the image after the first iteration. In this case we apply further passes of the algorithm until no more changes can be observed in the image. This is usually already the case after the second iteration, as shown in Figure 7.

4 Results

We have implemented the algorithm in a mixture of C# and MATLAB and ran it on an Intel Core2 Quad Q6600 @ 2.4 GHz, 8 GB RAM and Vista64 computer. Refer to Figure 6 for visual response.

Image	1	2	3	4
resolution	1140x1420	1802x1160	990x1400	548x884
3×3	4.3	8.5	12.2	3.1
11×11	41.7	61.2	87.8	34.1

Table 1: Comparison of the running times (given in seconds) for the first 4 images presented in Figure 6. The images were computed with a local neighborhood of 3×3 and 11×11 pixels over 5 iterations.

5 Discussion and Conclusion

Limitations. The optimization technique presented in section 3 does not always converge to an optimum. While this is usually not a big problem, the bad situation occurs when the symmetry neighborhood of a pixel contains two or more (roughly) equally balanced clusters. In this case it is not guaranteed, that the algorithm converges to the right configuration. Furthermore, neighboring pixels might converge at different clusters, which yields strong artifacts in the image, as shown in Figure 6 on the bottom right (the upper story of the building). We are currently working on an improved optimization method for our algorithm as well as on a more flexible symmetry detection strategy.

Conclusion. In this paper we presented a method to remove irregularities in a single approximately orthographic façade image using a symmetry propagation process. The symmetry is first detected using Monte Carlo sampling and encoded in a symmetry neighborhood. The symmetry is then propagated while performing edge-preserving smoothing on the image. This method can remove unwanted features, such as traffic lights, cables, signs and cars that are typically present in a façade image. It is not necessary to manually segment the unwanted image elements prior to running the algorithm, except for providing a coarse region mask. The output is intended to serve as input to rendering pipelines such as, e.g., Ali *et al.* [2]. We believe that our work is a useful solution to an important image processing step necessary in urban reconstruction projects.

Acknowledgments

We would like to acknowledge financial support from WWTF project number CI06025 as well as from NSF and NGA. Finally, we would like to acknowledge the Aardvark-Team in Vienna and the anonymous reviewers.

References

[1] Fernando Alegre and Frank Dellaert. A probabilistic approach to the semantic interpretation of building facades. Technical report, Georgia Institute of Technology, 2004.

[2] Saif Ali, Jieping Ye, Anshuman Razdan, and Peter Wonka. Compressed facade displacement maps. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):262–273, 2009.

[3] D. G. Bailey. Detecting regular patterns using frequency domain self-filtering. In *ICIP 1997, Set-Volume 1*, Washington, DC, USA, 1997. IEEE Computer Society.

[4] Danny Barash. A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):844–847, June 2002.

[5] Marcelo Bertalmio, Guillermo Sapiro, and Coloma Ballester. Image inpainting, 2000.

[6] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65 vol. 2, 2005.

[7] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, February 2008.

[8] Dabov, Foi, Katkovnik, and Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095, 2007.

Figure 6: We show image pairs of input façades and the result of our symmetry propagation. The running times of the first four examples are reported in Table 1. Note that the running time depends not only on the size of the image, but also on the number of symmetry neighbors and the degree of distortion. On images with a large symmetry neighborhood the running time takes up to several minuets. As an example observe image #3, whose running time is shorter than, e.g., image #2 in spite of the former's lower resolution. This is due to the quite large symmetry neighborhood of the façade and the variable number of iterations of the outlier rejection routine. The last example, on the bottom right, depicts a failure case of our algorithm. In the upper story the outlier rejection method could not determine the actual wall color coherently. (High-resolution version of this Figure is included in the supplemental material of the paper.)

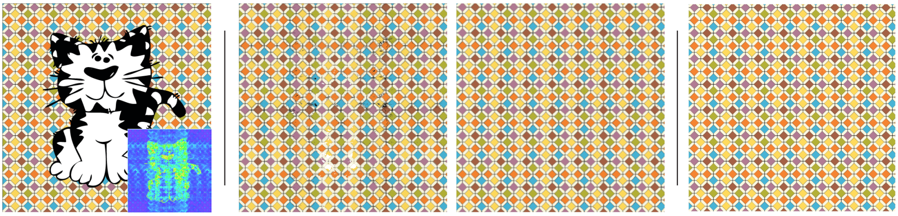

Figure 7: Left: An artificial test image with high symmetry (with a symmetry confidence map). Middle: The distortion has been removed in two iterations. Right: The ground truth.

[9] F. Han and S. C. Zhu. Bottom-up/top-down image parsing with attribute grammar. Technical report, January 2005.

[10] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[11] James Hays, Marius Leordeanu, Alexei Efros, and Yanxi Liu. Discovering texture regularity as a higher-order correspondence problem. pages 522–535. 2006.

[12] T. Korah and C. Rasmussen. 2d lattice extraction from structured environments. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 2, pages II – 61–II – 64, 2007.

[13] Thommen Korah and Christopher Rasmussen. Analysis of building textures for reconstructing partially occluded facades. In *European Conference on Computer Vision*, 2008.

[14] Yanxi Liu, T. Belkina, J. H. Hays, and R. Lublinerman. Image de-fencing. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[15] Yanxi Liu, R. T. Collins, and Y. Tsin. A computational model for periodic pattern perception based on frieze and wallpaper groups. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(3):354–371, 2004.

[16] Yanxi Liu, James Hays, Ying Q. Xu, and Heung Y. Shum. Digital papercutting. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, New York, NY, USA, 2005. ACM.

[17] Yanxi Liu, Wen C. Lin, and James Hays. Near-regular texture analysis and manipulation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 368–376, New York, NY, USA, 2004. ACM.

[18] Gareth Loy and Jan-Olof Eklundh. Detecting symmetry and symmetric constellations of features. In *In ECCV*, volume 2, 2006.

[19] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 560–568, New York, NY, USA, 2006. ACM.

[20] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. Symmetrization. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 63, New York, NY, USA, 2007. ACM.

[21] Pascal Müller, Gang Zeng, Peter Wonka, and Luc Van Gool. Image-based procedural modeling of facades. *ACM Trans. Graph.*, 26(3), July 2007.

[22] Manuel M. Oliveira, Brian Bowen, Richard Mckenna, and Yu sung Chang. Fast digital image inpainting. In *Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001*, pages 261–266. ACTA Press, 2001.

[23] Mark Pauly, Niloy J. Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J. Guibas. Discovering structural regularity in 3d geometry. volume 27, pages 1–11, New York, NY, USA, 2008. ACM.

[24] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990.

[25] Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz, and Thomas Funkhouser. A planar-reflective symmetry transform for 3d shapes. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 549–559, New York, NY, USA, 2006. ACM Press.

[26] Daniel Reisfeld, Haim Wolfson, and Yehezkel Yeshurun. Context free attentional operators: The generalized symmetry transform. *International Journal of Computer Vision*, 14:119–130, 1995.

[27] Jian Sun, Lu Yuan, Jiaya Jia, and Heung Y. Shum. Image completion with structure propagation. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 861–868, New York, NY, USA, 2005. ACM.

[28] Alexandru Telea. An image inpainting technique based on the fast marching method. *journal of graphics tools*, 9(1):23–34, 2004.

[29] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, Washington, DC, USA, 1998. IEEE Computer Society.