DOI: 10.1111/j.1467-8659.2011.01977.x
Eurographics Symposium on Rendering 2011
Ravi Ramamoorthi and Erik Reinhard
(Guest Editors)
Volume 30 (2011), Number 4

Stone Weathering in a Photograph

Su Xue Julie Dorsey Holly Rushmeier

Computer Graphics Group, Yale University

Figure 1: *Simulation of the progressive weathering of a tombstone in an input photograph (leftmost).*

Abstract

The appearance of weathering effects on stone is important for creating outdoor scenes in computer graphics. To achieve them, previous research has built upon physical simulation, which, while yielding a degree of realism, is computationally expensive and inapplicable to the situation when the object geometry is unknown. Also, physical simulation requires specific knowledge of the stone properties and environmental processes. In this paper, we present a simple visual simulation pipeline for creating weathering effects on stone within a single image. Two primary effects of stone weathering, i.e., smoothing *and* roughening*, are considered. In addition, erosion on the object silhouette is treated. These challenging effects involve significant geometry changes, which are intractable for previous image-based editing techniques. The effectiveness of our technique is illustrated on a variety of scenes and types of stone. While it can be fully automatic, it also allows easy user interaction.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithm; I.4.10 [Image Processing and Computer Vision]: Image Representation—Statistical.

1. Introduction

> *An important part of our culture is chiseled in stone, and we are in danger of losing it.*
>
> *S.Z.Lewin*

As stone is one of the most widely used materials in the man-made world, modeling and rendering weathering effects on stone is a recurring problem in computer graphics. In this paper, we explore the problem of visually simulating stone weathering within a single image.

While many previous techniques that model and render weathered stone have been developed based on physical simulation [MKM89, DEJ*99, KBKŠ09, JFBB10], none of them are applicable for images in the absence of geometry and specific stone properties. They are also too time-consuming for light-weight weathering applications, e.g., an image or a still object seen from a static viewpoint.

Several general weathering simulation techniques are also available, such as physically-based methods [CXW*05, MG08], and data-driven methods [LGR*05, GTR*06, LGG*07]. However, they all need geometry information, and some require tedious data acquisition, which are not applicable for weathering simulation within a single image.

On the other hand, image-based material editing – which allows the replacement of object appearance within an im-

Figure 2: *The overview of our system pipeline.*

age with new textures [FH04a] or new BRDFs [KRFB06] – has received increasing attention. However, these techniques do not address how to specifically simulate weathered appearance. Also, these techniques depend on geometry estimation, either shape-from-shading [FH04a] or dark-is-deep [KRFB06], which are prone to failure. Wang et al. [WTL*06] applied appearance manifolds to image-based material weathering. However, the shading variation within an image violates their assumptions and thus leads to failure. In subsequent work, Xue et al. [XWT*08] alleviated the affects of shading, however, the technique is still limited to the weathering effects rendered as pixel-wise color changes.

In contrast, stone weathering exhibits much more challenging effects than simple color changes, including significant geometry changes. Two primary weathering effects on stone are usually observed, i.e., *smoothing* and *roughening*. Smoothing is the wear of large-scale geometry, while stone texture remains, referred as *corestone weathering* in [DEJ*99]. This effect is well illustrated on the left tombstone in Fig. 3(a). Smoothing results from the gradual interaction between stone and physical or chemical forces, e.g., abrasion by sand particles carried by wind, or solution and advection by water running over the stone surface [Pri95, Cam95]. On the other hand, roughening is the falling off of chunks or flakes due to weaknesses in the material, which is caused by effects such as chemical decomposition or freeze-thaw cycles of water [AF83, Cam95]. The weathering on the shoulders and the chest of the statue in Fig. 3(b) illustrate this effect. Often, smoothing and roughening coexist, as shown on the flower relief in Fig. 3(a) and the face in Fig. 3(b). Other types of stone weathering effects that mainly involve color changes, such as biological growth and chemical deposition, could be dealt with by [WTL*06] and thus are not considered.

In this paper, we proposed a simple visual simulation pipeline for these challenging weathering effects on stone. An example result is shown in Fig. 1. It does not rely on physical priors or geometry estimation. The assumption is that the objects we simulate on are made of a *single type of stone*. A *segmentation-based editing framework* is adopted throughout the whole pipeline. Smoothing and roughening

Figure 3: *Photographs of stone weathering. (a) shows two similar tombstones at different weathering stages. (b) shows a stone statue before and after years of weathering. Figure (b) is reproduced from Fig. 1 in [Win87],* © *Springer-Verlag NY 1987. Included here by permission.*

effects are simulated in two sequential passes, preceding a final pass that deals with silhouette erosion. The pipeline is automatic and able to generate visually-plausible results for a wide variety of scenes and types of stone.

Our key contribution is that we for the first time simulate the weathering effects with significant geometry changes, such as smoothing and roughening effects, within a single image, which is not possible by previous methods. Though we focus on stone weathering, the segmentation-based editing framework and the techniques used in simulating smoothing effect, roughening effect, and silhouette erosion are very general for other image editing tasks. The technical details will be described in the following sections.

2. Overview

Fig. 2 shows our system pipeline that consists of four passes:

Patch Segmentation For each pixel $\mathbf{x}$ within an input image $I^{\dagger}$, a weathering tendency $d(\mathbf{x})$ is computed. Segmenting I based on $d(\mathbf{x})$ yields a set of non-overlapping patches, $\{p_i\}$. In the subsequent passes, we use p_i as the basic unit of editing, which is the segmentation-based editing framework.

Smoothing We simulate the smoothing effect on every patch, generating the result saved as S.

† Hereafter, I refers to the object(s) of interest that are manually masked.

Roughening We simulate the roughening effect over S. The result is saved as R.

Silhouette Erosion A *spheroidal erosion* technique is used to treat silhouette erosion of the objects. After a contrast adjustment, we achieve the final result.

2.1. Patch Segmentation

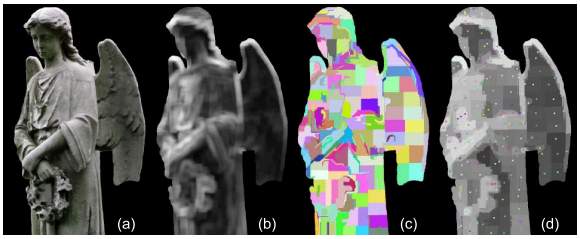


Figure 4: *Patch segmentation. (a) input I, (b) weathering tendencies $d(\boldsymbol{x})$, (c) the segmented patches $\{p_i\}$, (d) centroid pixels and average weathering tendencies of all patches.*

We used a *segmentation-based editing framework* inspired by high-level image editing tools [KSE*03, FH07, BSFG09], where they are dealing with applications like image completion, retargeting and texture synthesis. For our goal of weathering, we cluster the pixels sharing close *weathering tendencies* into one patch. As [WNH97] illustrates, the surface locations that have rich geometry features and intensive normal variations are more prone to weathering. In an image, these locations tend to exhibit higher variations in terms of pixel intensities, which is hence regarded as a robust measurement of weathering tendency.

For each pixel $\mathbf{x}$, we collect the intensities[‡] of the pixels within its square neighborhood of size $2r$. The variances of the intensities in *rgb* channels are computed respectively, which are summed as $v(\mathbf{x})$. The weathering tendency is then computed by $d(\mathbf{x}) = \left(\frac{v(\mathbf{x})}{v^*}\right)^{\eta}$, where $v^* = \max_{\mathbf{x}} v(\mathbf{x})$ and η is used to adjust the contrast in weathering tendencies. An example $d(\mathbf{x})$ is shown in Fig. 4(b). We performed a segmentation over I according to $d(\mathbf{x})$, yielding the patches $\{p_i\}$, each of which is a small area with similar weathering tendencies (see Fig. 4(c)). While the choice of segmentation methods is arbitrary, we use a fast method [FH04b] and refine its results by iteratively bisecting oversized patches until they are eliminated.

Parameter Setting As a smaller r favors faster computation, r is required to be larger than the scale of stone texture. We set $r = \sqrt{N(I)}/m$, where $N(I)$ is the number of pixels in the mask and hence $\sqrt{N(I)}$ a measure of the size of the input. In the segmentation method [FH04b], K and Ω are two key parameters, where a bigger K favors larger patches, and Ω is the number of pixels in the smallest patch allowed. Here we set $\Omega = N(I)/n$. In sum, η, m, K and n are primary parameters in our whole pipeline. We preset $\eta = 0.6$, $m = 60$, $K = 100$ and $n = 3200$, which automatically work well for various objects but also allow adjustment for better results. The summary of parameter settings for all presented results is shown in Table 1.

[‡] Pixel intensities take values within [0,1] in *rgb* colorspace. For LDR input, we first adjust its pixel intensities by a $\gamma = 2.2$ correction.

3. Smoothing Effect

3.1. Why Not Simply Blur I?

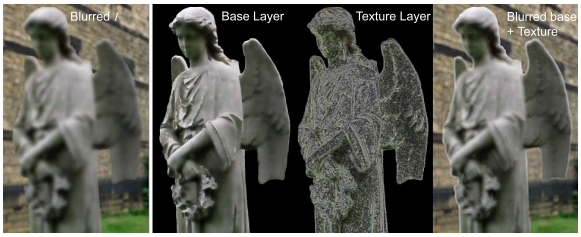


Figure 5: *Failure of straightforward blurring or blurring after "base+texture" separation.*

The smoothing effect is characterized by the significant loss of material and the preservation of stone textures. We experimented with Gaussian filtering over I as a straightforward method to blur surface geometries, however, stone textures and shading variations are also destroyed, as seen in the leftmost result in Fig. 5. We also tried using bilateral filtering [PD09] to separate I into the base and texture layers, diffusing the base layer and adding the texture layer back. However, the result with blurry artifacts fails to create the visual effects of weathering (see the rightmost result in Fig. 5), let alone the longstanding difficulty in dealing with the halo artifacts when combining different layers [BPD06]. These experiments show that achieving the two contradicting effects in smoothing, i.e., removing geometries and preserving realistic textures at the same time, is non-trivial.

3.2. Patch Updating

Based on the key observation that the original image usually contains the appearances of different weathering tendencies, we simulate the smoothing effect by updating the patches of higher tendencies with their matched peers of lower tendencies. As we assume that the objects are made of a single type of stone, the stone textures, assumed having finest-scale, are uniformly carried by the material. Consequently, the dominating features in computing weathering tendencies are distinguishable larger-scale geometry features. According to this insight, *patch updating* eventually updates the

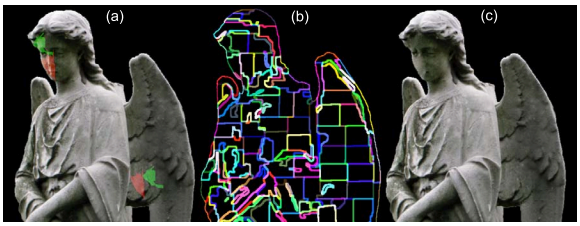


Figure 6: *Patch updating. (a) Two patches on the face and their matched peers. (b) The seams of segmentation preserve the boundary conditions. (c) The result of patch updating.*

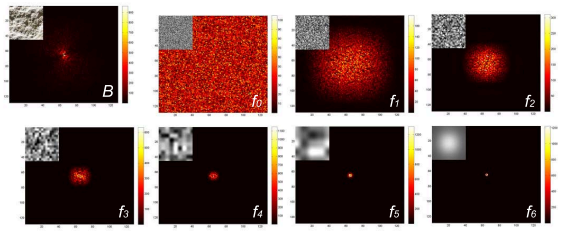


Figure 7: *The sample image (top-left) and the series of noise functions in both space and frequency domains.*

patches of richer geometry features with those of fewer geometry features, while stone textures are preserved as the original material is used. An example of matched pairs of patches are shown in Fig. 6(a).

Formally, for a patch p_0, we first compute its centroid pixel c_0 (Fig. 4(d)). Then for another pixel c, a patch p, which is centered at c and has the same shape as p_0, could be constructed. We search all pixels in I, finding the pixel c_1, where the corresponding patch p_1 minimizes the function $V_{p_0}(p)$:

$$V_{p_0}(p) = \|\bar{I}(p) - \bar{I}(p_0)\| + \|N(p) - N(p_0)\|/N(p_0) + w_d \bar{d}(p) \quad (1)$$

where $\bar{I}(p)$ is the average pixel intensities in patch p, $N(p)$ is the number of the in-mask pixels in p, and $\bar{d}(p)$ is the average weathering tendency in p. The weight w_d is set to 1.0 while adjustable. We accelerate the search using a hierarchical structure: start a full search by an interval of 2^{ξ}, $\xi = \max(\lceil \log_2 \sqrt{N(I)} \rceil - 4, 0)$; and then a subsequent search by an interval of $2^{\xi-1}$ in a square window of size 2^{ξ}, which is centered at the found location in the previous search. It iteratively proceeds until $\xi = 0$.

Now we are able to update p_0 using p_1, which has fewer geometry features but carries stone textures. The stage of weathering is described using a "time" parameter, t. Assume that the input image is at $t = 0$. When $t = \infty$, p_0 will be completely weathered, i.e., $I^{(\infty)}(p_0) = I^{(0)}(p_1)$. At a finite time t, $p_0^{(t)}$ has the appearance that is a weighted interpolation between $I^{(0)}(p_0)$ and $I^{(0)}(p_1)$. To preserve the shading distribution in I, we conducted the interpolation in the Laplacian field by setting

$$\Delta I^{(t)}(p_0) = w_{p_0} \Delta I^{(0)}(p_0) + (1 - w_{p_0}) \Delta I^{(0)}(p_1) \quad (2)$$

where Δ is the spatial Laplacian operator; $w_{p_0} = e^{-k\bar{d}(p_0)t}$, $k = 0.01$. $I^{(t)}(p_0)$ is therefore solved as a Poisson equation [PGB03] with the boundary condition $I^{(t)}(\partial p_0) = I^{(0)}(\partial p_0)$, where ∂p_0 are the boundary pixels of p_0. The boundary pixels of patches and the updating result are illustrated in Fig. 6(b) and (c).

This way, we are able to generate a sequence of consistently weathered objects at different times. Fig. 2 shows a result of smoothing, saved as S. The power of the segmentation-based editing framework emerges: despite the fact p_1 probably mismatches p_0 in terms of normal orientations, this mismatch is invisible on the patch scale as the overall shading distribution is preserved.

4. Roughening Effect

4.1. Simulating Rough Appearance

While we reuse the original patches to achieve the smoothing effect, the roughening effect can not be simulated this way. This effect is characterized by the rough appearance which is very different from the normal stone appearance and is not necessarily present in the original image.

Noticing that the rough appearance of stone exhibits a strong self-similarity in its geometry, physicists have used stochastic models and fractal models to describe it [JFG*03, AADM97]. Here we propose to learn a fractal function derived from photographic examples. This function modulates the target image and thus visually reproduces rough appearance. Inspired by the success of fractal in modeling terrains [MKM89], we generate the 2D fractal function $F(\mathbf{x})$ by adding up a series of band-limited noises with successive frequencies,

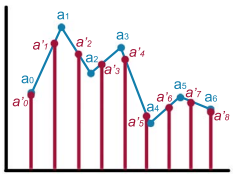
$$F(\mathbf{x}) = \sum_{i=0}^{\mathcal{L}} a_i f_i(\mathbf{x}) \quad \text{s.t.,} \quad a_i \geq 0 \quad (3)$$

where $f_i(\mathbf{x})$ is a 2D noise function, drawing values from a standard normal distribution at the grid points spaced by an interval of 2^i. The values at in-between locations are cosine-interpolated. The number of noise functions we used is $\mathcal{L} = \lfloor \log_2(\max(w,h) - 1) \rfloor$, where w,h are the width and height of the target image to be modulated.

An ordinary $1/f$ noise sets $a_i = 2^i$. However, it does not reflect the specific properties of the rough appearance of stone. To this end, we conduct a simple pre-computation phase, learning a_i from a sample image $B(\mathbf{x})$ of rough stone surface. After transforming $B(\mathbf{x})$ and $f_i(\mathbf{x})$ into the frequency domain, we compute a_i by minimizing

$$\left| \sum_{\omega} \|\tilde{B}(\omega)\|^2 - \sum_{i=0}^{\mathcal{L}_B} \sum_{\omega} \|a_i \tilde{f}_i(\omega)\|^2 \right| \quad \text{s.t.,} \quad a_i \geq 0 \quad (4)$$

where $\tilde{B}(\omega), \tilde{f}_i(\omega)$ are their Fourier transforms, ω is a 2D frequency, $\mathcal{L}_B = \lfloor \log_2(\max(w_B, h_B) - 1) \rfloor$, w_B, h_B are the width and height of the sample. Fig. 7 illustrates the sample image and the noise functions in both domains.



Since the size of the sample and that of the target do not necessarily match ($\mathcal{L} \neq \mathcal{L}_B$), we compute the coefficients $a'_i, i = 0..\mathcal{L}$ for generating $F(\mathbf{x})$ by interpolating learned $a_i, i = 0..\mathcal{L}_B$, as shown in the inset. $F(\mathbf{x})$ is then used to modulate the luminance channel of the smoothing result S, i.e., L_S. Formally, $L_S(\mathbf{x}) \leftarrow L_S(\mathbf{x}) + F(\mathbf{x})$. The scale of a'_i does not matter because we scale new L_S so that its range equals the range of the original L_S.

Despite the fact that various types of stone exhibit slightly different appearance in roughening, we experimentally observed that our sample works plausibly on various examples, which reveals that rough appearance is transferrable for the purpose of visual simulation, though, other samples could also be used for variety. The examples of our simulated rough appearance on different types of stone are shown in Fig. 8. We compare the roughness of the reproduced appearance and that of the sample image by computing their fractal dimensions [KCC89], which are consistent.

Figure 8: *Simulating rough appearance on various types of stone.*

4.2. Applying Rough Appearance

In nature, stone exhibits roughening at a few locations. Although we allow users to specify the sites for roughening through interaction, by default we provide an automatic detection method. $\beta(\mathbf{x}) = (d(\mathbf{x}))^{\nu}$ is used as the probability of $\mathbf{x}$ being such a site, where ν is set to 3.0. Similar to smoothing, the modulation by $F(\mathbf{x})$ is controlled by the time parameter t and $\beta(\mathbf{x})$, i.e., $L_S(\mathbf{x}) \leftarrow L_S(\mathbf{x}) + w_r F(\mathbf{x})$, where $w_r = e^{-k\beta(\mathbf{x})t}$, $k = 0.01$. The result that has gone through both smoothing and roughening is shown as R in Fig. 2.

5. Silhouette Erosion

In the last step, we introduce a *spheroidal erosion* technique to treat the silhouette erosion of objects in R, while avoiding the "cutout" artifact. While smoothing and roughening simulate the weathering effects on surface appearance and geometry, silhouette erosion simulates the loss of materials on

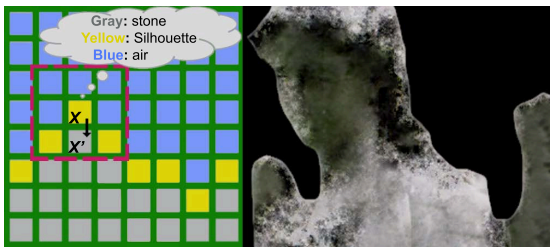


Figure 9: *Left: spheroidal erosion on silhouette. Right: natural lighting is preserved on silhouette after erosion.*

weathered silhouettes. A silhouette pixel is defined as having at least one 4-connected neighbor that is out of the object.

As [JFBB10] points out, the sharper corners of stone tend to be rounded faster, which is termed "spheroidal weathering". In their simulation, the spheroidal weathering rate at a given point is proportional to its positive curvature, which is computed by the ratio of rock to air in a fixed volume. We adapt this idea to our 2D spheroidal erosion of silhouette. At a silhouette pixel $\mathbf{x}$, its erosion rate $\rho(\mathbf{x})$ is computed by the ratio of stone to air in a 7×7 square neighborhood

$$\rho(\mathbf{x}) = w_a N_a / (w_a N_a + w_s N_s) \tag{5}$$

where N_a, N_s are the numbers of air and stone pixels in the neighborhood, w_a, w_s are their corresponding weights, both normally set to 1.0. To simulate the phenomenon that the stone exposed to upper air is more prone to weathering, we assign 2.0 to the weights of the air pixels above $\mathbf{x}$. Fig. 9 illustrates this scheme. We initialize all silhouette pixels with the same durabilities $\delta_0(\mathbf{x})$, which are decreased at their weathering rates in each simulation iteration, $\delta_{n+1}(\mathbf{x}) = \delta_n(\mathbf{x}) - \rho(\mathbf{x})$. When $\delta_n(\mathbf{x})$ falls below zero, $\mathbf{x}$ is completely "eroded" and thus removed from the object. The locations of silhouette pixels and their weathering rates are recomputed in every iteration. The number of iterations is determined by $\varepsilon\sqrt{N(I)}t$, where ε is an adjustable speed parameter set to 0.0002 in our experiments.

However, this scheme results in the artifact that the new formed silhouette is unnaturally sharp, like a paper cutout. Noticing that the natural silhouette usually exhibits environment lighting, e.g., back-lighting, we propagate the colors of the original silhouette inward when silhouette erosion happens. Algorithmically, if the durability of a silhouette pixel $\mathbf{x}$ is to reach zero in the current iteration, its intensity is first bled into each of its neighboring non-silhouette pixel $\mathbf{x}'$, i.e., $R(\mathbf{x}') \leftarrow \alpha R(\mathbf{x}') + (1 - \alpha) R(\mathbf{x})$, $\alpha = 0.2$ in practice. This way, the natural lighting conditions on the original silhouette are protected after silhouette erosion, as seen on the silhouette pixels in the right figure of Fig. 9

Contrast Adjustment Finally, the luminance histogram of the result is adjusted to match that of the original image [HB95], by which the deviation in contrast after the whole simulation pipeline is avoided.

Figure 10: *A sandstone monkey.*

Figure 11: *A granite statue of Pharaoh.*

Figure 12: *A marble column.*

6. Results

Here we present a wide variety of examples to illustrate our techniques. In the results, the weathered objects are pasted back onto the backgrounds. The missing areas in the background due to silhouette erosion are manually inpainted. An automatic inpainting technique could also be used. All simulations were run on a Intel® Core2 Quad 2.83GHz machine using a single thread on CPU. For our experimental images of resolutions ranging from 1M to 5M pixels, the computation time varies from 1 to 10 minutes.

In Fig. 2, we have shown the weathering effects on a stone angel statue with complex geometry under challenging shading conditions.

Fig. 1 and Fig. 10 show dramatic weathering effects on the types of stone with rough textures and fragile durabilities. Our smoothing technique wears away geometry features, while nicely saving stone textures.

Fig. 11 shows the result on granite, which has rich grains and strong durability. Its slow weathering process is subtly revealed via the worn edges of beard, lips, ears and hat. Fig. 12 demonstrate the results on marble, which also has high hardness.

Fig. 13 and Fig. 14 are two examples of the effacing of relief sculptures.

Finally, we simulate the weathering process of the less-weathered objects in Fig. 3 (a) and (b) as a validation. The results are shown in Fig. 15 and Fig. 16, which are consistent and visually plausible.

The supplementary videos show the evolving animations of the examples above. They demonstrate the ability of our simulation to create consistently varying appearance of weathered stone. The summary of parameter settings for all examples is listed in Table 1. For some examples, e.g., Fig. 1,13, 14,15, silhouette erosion was turned off.

Limitations

Besides the assumption that the scale of stone textures is distinguishably smaller than that of geometry features, successful smoothing also requires that the variance of geometry scales is not dramatic. Otherwise, an inappropriate weathering tendency map will lead to a failure in segmentation. This explains why our technique mainly works for stone "objects", rather than stone buildings, which often include

Figure 13: *A calendar carved on stone.*

Figure 14: *A relief sculpture.*

Figure 15: *Simulation on the image in Fig.3(a).*

Figure 16: *Simulation on the image in Fig.3(b).*

objects of dramatically varying scales. As shown in Fig. 17, though the relief sculptures and the columns, which have relatively small feature scales, are eroded, the whole building does not look correctly weathered at its large scale.

7. Conclusion

In this paper, we propose a simple visual simulation pipeline for stone weathering within a single image. The absence of physical priors and geometry information precludes the application of previous physical simulation methods. In addition, existing image editing techniques are unable to deal with the significant geometry changes caused by stone weathering. Our method achieves visually-plausible simulations of these challenging effects on a variety of types of stones.

Acknowledgement

The authors thank the anonymous reviewers for their comments. Photos *monkey* and *temple* are courtesy of Digital Collections of Yale University Library. We would like to thank *Adam Hart-Davis (angel), Marian Ware (calendar), Robert Potvin (tombstone), Don Sutherland (pharaoh), John Roumeliotis (column), Rick Jacobi (relief)* for allowing us to use their photos.

References

[AADM97] ARRAULT J., ARNÉODO A., DAVIS A., MARSHAK A.: Wavelet based multifractal analysis of rough surfaces: Application to cloud models and satellite data. *Phys. Rev. Lett. 79* (1997), 75–78. 4

[AF83] AMOROSO G. G., FASSINA V.: *Stone Decay and Conservation*. Elsevier, New York, NY, 1983. 2

	Angel	Monkey	Pharaoh	Column	Relief	Calendar	Tombstone	Fig.3(a)	Fig.3(b)	Temple
η	0.4	0.4	0.8	0.6	0.6	0.6	0.4	0.5	0.6	0.3
m	40	40	60	100	100	100	100	100	60	100
K	100	100	50	100	100	100	100	100	100	25
n	1600	3000	3200	1600	1600	3200	1600	3200	1600	4800
w_d	1.0	0.8	1.1	0.8	1.2	1.2	1.2	1.2	0.1	0.4

Table 1: *Parameter settings of the presented examples.*

Figure 17: *Failure case. A temple that includes objects of dramatically varying scales.*

[BPD06] BAE S., PARIS S., DURAND F.: Two-scale tone management for photographic look. *ACM Trans. Graph. 25* (July 2006), 637–645. 3

[BSFG09] BARNES C., SHECHTMAN E., FINKELSTEIN A., GOLDMAN D. B.: Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph. 28* (July 2009), 24:1–24:11. 3

[Cam95] CAMUFFO D.: Physical weathering of stones. *Science of The Total Environment 167* (May 1995), 1–14. 2

[CXW*05] CHEN Y., XIA L., WONG T.-T., TONG X., BAO H., GUO B., SHUM H.-Y.: Visual simulation of weathering by gamma-ton tracing. *ACM Trans. Graph. 24* (July 2005), 1127–1133. 1

[DEJ*99] DORSEY J., EDELMAN A., JENSEN H. W., LEGAKIS J., PEDERSEN H. K.: Modeling and rendering of weathered stone. In *Proc. of SIGGRAPH '99* (1999), pp. 225–234. 1, 2

[FH04a] FANG H., HART J. C.: Textureshop: texture synthesis as a photograph editing tool. *ACM Trans. Graph. 23* (August 2004), 354–359. 2

[FH04b] FELZENSZWALB P. F., HUTTENLOCHER D. P.: Efficient graph-based image segmentation. *Int. J. Comput. Vision 59* (September 2004), 167–181. 3

[FH07] FANG H., HART J. C.: Detail preserving shape deformation in image editing. *ACM Trans. Graph. 26* (July 2007). 3

[GTR*06] GU J., TU C.-I., RAMAMOORTHI R., BELHUMEUR P., MATUSIK W., NAYAR S.: Time-varying surface appearance: acquisition, modeling and rendering. *ACM Trans. Graph. 25* (July 2006), 762–771. 1

[HB95] HEEGER D. J., BERGEN J. R.: Pyramid-based texture analysis/synthesis. In *Proc. of SIGGRAPH '95* (1995), pp. 229–238. 5

[JFBB10] JONES M. D., FARLEY M., BUTLER J., BEARDALL M.: Directable weathering of concave rock using curvature estimation. *IEEE Trans. on Visualization and Computer Graphics 16* (January 2010), 81–94. 1, 5

[JFG*03] JAFARI G. R., FAZELI S. M., GHASEMI F., VAEZ ALLAEI S. M., REZA RAHIMI TABAR M., IRAJI ZAD A.AND KAVEI G.: Stochastic analysis and regeneration of rough surfaces. *Phys. Rev. Lett. 91* (2003), 226101–226104. 4

[KBKŠ09] KRIŠTOF P., BENEŠ B., KŘIVÁNEK J., ŠT'AVA O.: Hydraulic erosion using smoothed particle hydrodynamics. *Computer Graphics Forum 28* (2009), 219–228. 1

[KCC89] KELLER J. M., CHEN S., CROWNOVER R. M.: Texture description and segmentation through fractal geometry. *Comput. Vision Graph. Image Process. 45* (February 1989), 150–166. 5

[KRFB06] KHAN E. A., REINHARD E., FLEMING R. W., BÜLTHOFF H. H.: Image-based material editing. *ACM Trans. Graph. 25* (July 2006), 654–663. 2

[KSE*03] KWATRA V., SCHÖDL A., ESSA I., TURK G., BOBICK A.: Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph. 22* (July 2003), 277–286. 3

[LGG*07] LU J., GEORGHIADES A. S., GLASER A., WU H., WEI L.-Y., GUO B., DORSEY J., RUSHMEIER H.: Context-aware textures. *ACM Trans. Graph. 26*, 1 (2007). 1

[LGR*05] LU J., GEORGHIADES A. S., RUSHMEIER H., DORSEY J., XU C.: Synthesis of material drying history: Phenomenon modeling, transferring and rendering. In *EGNP* (2005), pp. 7–16. 1

[MG08] MÉRILLOU S., GHAZANFARPOUR D.: A survey of aging and weathering phenomena in computer graphics. *Computers and Graphics 32*, 2 (2008), 159 – 174. 1

[MKM89] MUSGRAVE F. K., KOLB C. E., MACE R. S.: The synthesis and rendering of eroded fractal terrains. In *Computer Graphics Proc.* (1989), SIGGRAPH '89, pp. 41–50. 1, 4

[PD09] PARIS S., DURAND F.: A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vision 81* (January 2009), 24–52. 3

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Trans. Graph. 22* (July 2003), 313–318. 4

[Pri95] PRICE D. G.: Weathering and weathering processes. *Quarterly Journal of Engineering Geology 16* (June 1995), 243–252. 2

[Win87] WINKLER E.: Weathering and weathering rates of natural stone. *Environmental Geology 9* (1987), 85–92. 10.1007/BF02449939. 2

[WNH97] WONG T.-T., NG W.-Y., HENG P.-A.: A geometry dependent texture generation framework for simulating surface imperfections. In *Proc. of the Eurographics Workshop on Rendering Techniques '97* (1997), pp. 139–150. 3

[WTL*06] WANG J., TONG X., LIN S., PAN M., WANG C., BAO H., GUO B., SHUM H.-Y.: Appearance manifolds for modeling time-variant appearance of materials. *ACM Trans. Graph. 25* (July 2006), 754–761. 2

[XWT*08] XUE S., WANG J., TONG X., DAI Q., , GUO B.: Image-based material weathering. *Computer Graphics Forum 27* (2008), 617–626. 2