Abstract

This paper proposes an interactive approach using joint image-noise filtering for achieving high quality image-noise separation. The core of the system is our novel joint image-noise filter which operates in both image and noise domain, and can effectively separate noise from both high and low frequency image structures. A novel user interface is introduced, which allows the user to interact with both the image and the noise layer, and apply the filter adaptively and locally to achieve optimal results. A comprehensive and quantitative evaluation shows that our interactive system can significantly improve the initial image-noise separation results.

1 Introduction

Image noise is the digital equivalence of film grain found in analogue cameras. For digital images, noise appears as random speckles on an otherwise smooth surface. Image quality can be significantly degraded by noise, and thus tasks that require accurate image information will be adversely affected.

On the other hand, many photo-editing applications require noise to be preserved in the resulting images to make them look natural. In fact, adding noise after photo editing has become a standard feature in many professional image editing tools such as Adobe Photoshop [Adobe Systems 2008b].

In this paper we focus on the central problem of image-noise separation: decomposing an input noisy image $I$ into a noise-free latent layer $I_l$ and a noise layer $I_n$ as

$$I = I_l + I_n.$$  

(1)

For the purpose of denoising, $I_n$ can be simply discarded and $I_l$ is the final output. In other applications where the noise characteristics need to be preserved, various image editing operations can be applied on $I_l$ to produce a new latent image $I'_l$. $I_n$ can then be applied back to generate the final image as

$$I' = I'_l + f_n(I'_l, I_l, I_n),$$  

(2)

where $f_n$ is a noise synthesis operator (see Section 6.1 for details).

One way to achieve this decomposition is to apply existing noise reduction/removal algorithms which aim at estimating a clean latent image $I_l$. This problem has been extensively studied for many decades in signal and image processing, resulting in a large volume of research literature as well as commercial software packages. However, previous approaches have exclusively focused on automatic noise estimation and removal, i.e., a good estimation of the latent image. This is extremely challenging, since accurate modeling of image noise depends critically on a number of factors such as capturing instruments, data transmission media, image quantization, and discrete sources of radiation [Motwani et al. 2004]. Any single algorithm is unlikely to work well in all cases. We argue that two conflicting issues should be adequately addressed at the same time to achieve high quality image-noise decomposition:

**Over-smoothed image structures.** High-frequency image structures are often mistakenly estimated as noise and removed from
$I$, resulting in an over-smoothed $I_r$. We call the portion of image structure in the noise layer *structure residual*.

**Residual noise in smooth regions.** If the noise level in $I$ is underestimated, noise residual will appear in smooth regions of $I_r$.

Previous approaches typically optimize the latent image without adequate consideration of the quality of the noise layer. Although a set of parameters is usually provided which can be tuned to better adapt to the input image, they are insufficient to handle both artifacts especially over-smoothed image structures. Furthermore, once the high frequency image structures are over-smoothed, there is no existing good solution to recover them. As a result, using previous image denoising approaches alone often cannot produce the desired high quality image-noise separation.

In this paper, we present an interactive system to achieve high quality image-noise separation. Given an initial image-noise decomposition produced by an automatic method, our approach jointly considers the characteristics of image structures and pure noise, and aims at producing a noise-free latent image and a structure-free noise layer. To achieve this we propose a joint image-noise filter, which is more effective in separating noise from high-frequency and low-frequency image structures than existing edge-aware filters. Furthermore, we propose a novel user interface which allows the user to adaptively apply the filter on both the image and the noise layer for eliminating artifacts on both layers to achieve optimal results. This sets our approach apart from automatic approaches where the user is limited only to empirically tweak some global parameters.

We perform a comprehensive evaluation to demonstrate the effectiveness of our system. Finally, we show how our system can be employed in a variety of noise-consistent image editing tasks where unique noise characteristics can be preserved in the editing process.

## 2 Related Work

### Image Denoising Algorithms.

Natural images exhibit sparsity in the wavelet domain [Mallat 1989]. By decomposing a noisy image into subbands, the coefficients of the clean image can be estimated by per-pixel suppression of low-amplitude values [Simoncelli and Adelson 1996], or inferred using a joint prior involving image neighborhood [Dabov et al. 2007; Portilla et al. 2003].

Anisotropic diffusion [Perona and Malik 1990] works in the image domain, and iteratively performs edge preserving filtering. Bilateral filtering [Tomasi and Manduchi 1998] is a non-iterative alternative to anisotropic diffusion, which has been widely adopted in edge-preserving image editing tasks. Both techniques can be used for noise reduction.

Recently, Markov random fields [Roth and Black 2005; Weiss and Freeman 2007] and conditional random fields [Liu et al. 2008] have also been applied in denoising in computer vision. These approaches use different statistical image priors in a graph setting to infer the clean image.

Most of these approaches, however, do not consider the quality of the noise layer. In a few approaches [Buades et al. 2008], the extracted noise layer was shown for visual evaluation of the quality of the denoising result. However, to the best of our knowledge, no algorithm has actively utilized the noise layer to improve the denoising quality.

### User Controls.

Most commercial denoising software packages [ABSoft Inc. 2008; Imagenomic Inc. 2008; Adobe Systems 2008b] provide a number of parameters for the user to adjust in order to obtain a better result. For instance, the user can adjust the global noise level after it has been automatically estimated, the strength of the denoising operation, and other algorithm-specific parameters. Noise Ninja [PictureCode Inc. 2008] also provides the user with a local editing brush, but its functionality is limited only to undo or redo certain denoising operations once in a local region. In contrast, the user controls proposed in our system are more flexible and effective in assisting the user in generating high quality results.

### Noise Aware Image Editing.

Some existing image editing tools have limited capabilities to deal with noise. For instance the “threshold parameter” in Photoshop’s unsharp mask can suppress certain amount of noise when sharpening the image, which however cannot distinguish very well noise from high frequency image structures, as we will show in Section 6. Adding noise/grain is a common feature in imaging software [Adobe Systems 2008a; Adobe Systems 2008b], but the synthesized noise is derived from a presumed parametric model which may not match the desired noise characteristics. We show in this paper how our system can be used for achieving high quality noise-consistent image editing.

## 3 Joint Image-Noise Filtering

### 3.1 Noise Prior

Let us examine again the noise layer $I_n$ shown in Figure 1(b), which presents strong thin lines corresponding to the high-frequency structural edges. Although the structure residuals are visually noticeable, separating them from pure noise is not an easy task, since both are high frequency signals.

One important prior we assume is that noise, by definition, is spatially random. Although CCD imaging or film grain may cause neighboring pixels and their noise levels to be locally correlated, the correlation is limited to a very small spatial range (1 to 2 pixels typically), and the spatial randomness (ergodicity) of noise is still valid at larger scales. Since most of the computations in our system are carried out in relatively large local image windows (except the noise synthesis method described in Section 6.1), the effects of the small scale correlation can be safely ignored. In fact, the ergodicity of noise is the fundamental assumption in many existing image filtering methods ranging from anisotropic diffusion [Perona and Malik 1990] to the non-local means method [Buades et al. 2008].

Spatial randomness is easy to measure by correlation, but hard to optimize directly. We therefore adopt an alternative approach by further constraining our system to only consider additive noise with zero-mean, thus the “local average” (will be detailed later) of noise values in a pure noise layer should be zero. Hence, whenever the local average is not zero, it is assumed to be caused by structure residuals, which, unlike noise, present strong and large scale correlations. This correlation can be easily observed as shown in Figure 1(d). Based on this observation, instead of directly optimizing noise randomness, we in turn compute the expectation of the structure residual inherent in the noise. Once the structure residual $I_n$ is estimated, the pure noise signal $I_n$ can simply be given by $I_n = I - I_n$.

### 3.2 Extracting the Structure Residual

Based on our noise assumption, for a pixel $p$ in the noise layer, the expectation of its structure residual $I_n(p)$ can be estimated as the average of a large number of nearby noise samples, where the weight for each sample describes how likely this sample contributes to the same structure residual as $p$ does. The design of the weighting function is thus extremely important for an unbiased estimation of $I_n(p)$.
In our system, we assume that for a nearby pixel \( q \), it has a high probability of having the same residual as \( p \) if the following criteria are met:

1. \( p \) and \( q \) are spatially close to each other;
2. \( p \) and \( q \) have similar colors in the recovered image layer \( I_l \);
3. \( p \) and \( q \) are located on the same image structure.

Criterion 1 and 2 basically assume that image properties vary smoothly except across an edge. These two criteria have been extensively used in many edge-aware image filtering algorithms such as Bilateral filtering [Tomasi and Manduchi 1998] and Weighted Least Square (WLS) decomposition [Farbman et al. 2008]. Criterion 3 is unique in our system, which restricts the local averaging to be on the same object surface.

To better explain the importance of Criterion 3, we use an example of a local noise layer \( I_n \) shown in Figure 2(a), and Figure 2(b) and (c) are extracted structure residual layer \( I_s \) by different means. In Figure 2(a), \( E_A \) and \( E_B \) are obviously structure residuals of two image edges close to each other. Suppose we want to compute the expectation of the structure residual value for pixel \( p \) using the average of its nearby noise samples. If we do not enforce Criterion 3, then we essentially define an isotropic neighborhood region centered at \( p \) (the red curve in Figure 2(b)), thus samples on the nearby edge will be included in the estimation, even they should have lower weights according to Criterion 2. Since samples for computing \( E_A \) (resp. \( E_B \)) are drawn from different distributions, mixing them together will cause the estimation to be biased, so the structure residual cannot be adequately extracted as shown in Figure 2(b). On the contrary, employing Criterion 3 will ensure that we only sample pixels on the same structure as \( p \). Since these samples are likely to obey the same distribution as \( p \), an unbiased estimation can be achieved, leading to a more accurate structure residual extraction shown in Figure 2(c).

**Figure 2:** (a) Initial noise layer. (b) and (c) are extracted structure residuals using the isotropic kernel (red) and the structural-aware kernel (yellow), respectively.

We propose a joint image-noise filter which satisfies all the three criteria described above. This filter can be applied to the latent image layer, noise layer, or both. Mathematically, the structure residual at \( p \) is computed as:

\[
I_s(p) = \frac{1}{A} \sum_{p' \in A} I_l(p') - I_l(p) = USU^T,
\]

where \( A \) is the number of pixels in neighborhood \( A \).

In our system we set the radius of the neighborhood \( A \) to be 4 \( \sigma_s \). Note that weight \( W_s(p; q) \) is asymmetric on \( p \) and \( q \). We then define

\[
W_s(p, q) = \frac{1}{2} (W_s(p; q) + W_s(p; q))
\]

as a symmetric measurement of how likely \( p \) and \( q \) are on the same structure.

Figure 3 shows an example of applying the proposed filter for refining the noise layer. The synthetic input image (not shown) is corrupted by Gaussian noise in the luminance channel. We first apply a baseline denoising algorithm on the input image, which generates a noise layer containing strong structure residuals. By further applying joint image-noise filtering on the noise layer, we can effectively recover from the noise layer the high frequency structure residuals, resulting in a much sharper latent image as well as a better quality noise layer.

**Figure 3:** Using the joint image-noise filter for improving image-noise separation. Top: initial latent image and noise layer. Bottom: Improved latent image and noise layer after filtering. Note that the filtered latent image is much sharper than the initial latent image.

In Figure 2 we have motivated the importance of the structure term in achieving unbiased structure residual estimation. We now further demonstrate its significance using an example in Figure 4, where we compare the results generated by the filter with and without the structure term in Equation 6. The comparison clearly shows that with the structure term, we can estimate the structure residual more...
accurately. It is worth mentioning that without the structure term, the proposed filter will be reduced from one of structure-aware to edge-aware, which has a weighting scheme similar to the commonly used bilateral filter. The substantial improvement shown in this experiment indicates that high quality image-noise separation is not easily achievable by directly adapting existing edge-aware filters for joint image-noise filtering. We have observed similar phenomena consistently in numerous examples shown in this paper.

Figure 4: Comparison of joint image-noise filtering with and without the structure term proposed in Equation 6.

3.3 Joint Image-Noise Filtering vs. Image Filtering

It is both interesting and worthwhile to compare the proposed joint image-noise filtering algorithm against traditional image filtering approaches, i.e., algorithms that only optimize the latent image layer for noise removal. Image filtering can be regarded as a process of computing the expectation of true image signal from the input noisy image, without explicitly considering the property of the noise layer. Representative techniques in this category range from the basic isotropic Gaussian filtering to more advanced anisotropic diffusion, bilateral filtering and the nonlocal means method.

Although most advanced image filtering algorithms were designed to preserve sharp image edges while removing high frequency noise, we found that they can only partially achieve this goal. To demonstrate this phenomenon, we use a 1D example shown in Figure 5. The blue line in Figure 5(a) is the input noisy signal generated from the latent smooth signal shown in green. The red curve is the filtered signal using bilateral filtering, with the size of the spatial neighborhood shown in brown. Using bilateral filtering, although neighborhood data points with large value differences are assigned lower weights in the averaging process, the estimated signal around the structural edge are still biased due to the underlying signal change. This will result in structure residuals in the noise layer as shown in Figure 5(b). Reducing the range parameter used in the bilateral filter may reduce the structure residual in the noise layer, but more noise will remain in the filtered signal as well.

This problem can be largely resolved by applying the proposed joint signal-noise filtering. Figure 5(c) and (d) show the refined signal and noise after we extract the structure residual from Figure 5(b) using the joint filter, and add it back to the signal. Note that the refined signal (red curve) is now more faithful to the true latent signal in Figure 5(c), compared with the initial result in Figure 5(a). In essence, the high frequency structure residual in the image domain becomes the low frequency structure in the noise domain, which can be effectively separated from pure random noise.

Figure 5: 1D illustration of expectation bias in signal filtering (a) and (b). Observe the less biased result obtained after applying joint signal-noise filtering in (c) and (d).

In Figure 6, we compare various results generated by several representative edge-aware image filters, including the bilateral filter [Tomasi and Manduchi 1998], the WLS filter [Farbman et al. 2008], and the nonlocal means method [Buades et al. 2008]. These methods can produce relatively sharp and clean results with properly set parameters; however, unwanted structure residuals in the noise layer are still apparent. Note that although the concept of “method noise” was introduced in [Buades et al. 2008] for visual evaluation using the noise image, the noise image itself is not explicitly optimized. By further applying joint image-noise filtering after image filtering, remaining structure residuals can be effectively recovered.

Figure 6: Comparison of image-noise separation results.

3.4 Setting Parameters

The proposed joint image-noise filter has three major parameters $\sigma_s$, $\sigma_d$ and $\sigma_c$ (see Equation 4-6). Generally speaking, $\sigma_c$ is correlated to the noise level present in the input image, and $\sigma_d$ and $\sigma_s$ should be proportional to the size and curvature of local image structures. Although they can be tweaked independently, in practice we found that these parameters can be jointly increased or decreased to reflect the overall sensitivity of the filter. Furthermore, in our system we set $\sigma_s = \sigma_d$ as they both control the spatial sensitivity of the filter.
Intuitively, the parameters control the “minimal scale” of the structures that can be separated from noise by applying the filter. To better demonstrate this, in Figure 7 we apply our joint image-noise filter with three different sets of parameters on the noise image in Figure 4. As we can see, if the parameters are too small, the filter tends to extract image structures at all scales, thus noise signal is also mistakenly extracted as structure. On the other hand, if the parameters are too large, then only large scale structures can be extracted and smaller structures are left as noise. Setting the parameters properly can generate the optimal result in Figure 7(b) where image structures are largely removed while the noise remains unaffected.

This experiment demonstrates the importance of properly setting parameters for achieving high quality results. In the next section we will describe how to achieve this in an interactive fashion.

4 Interactive Image-Noise Separation

Although the joint image-noise filter can be applied automatically with fixed parameters, as shown in Figure 4, it can perform significantly better with locally adaptive parameters that can be actively controlled by the user. The pertinent results will be demonstrated quantitatively in Section 5. In this section, we illustrate how the filter can be applied in an interactive fashion for achieving high quality results. Our user interface is shown in Figure 9.

4.1 The Work Flow

We develop a two-step work flow to help the user achieve a good balance between efficiency and accuracy. The first step is global filtering. Given an initial image-noise decomposition as input, the joint image-noise filter is applied globally on the noise layer, using a set of fixed parameters chosen by the user. The user first applies the filter in a small region of the image in order to find the best settings. Once filtered, the extracted structure residual is subtracted from the noise layer and then added to the latent image. The same global filtering process can also be applied on the latent layer to remove from smooth regions noise residuals, which are then added back to the noise layer.

The image-noise separation result is usually largely improved after the global filtering step. However, the result will not be optimal since a fixed set of parameters may not work well for all local regions. The user then uses mouse brushes to apply local filtering to areas that need further improvement. Details of the user brushes will be described next. By employing the two-step approach, the user’s effort is minimized, as he/she only needs to touch up a small number of areas where artifacts remain after global filtering.

4.2 The Structure Brush

Our UI provides a structure brush for the user to recover high-frequency image structures which have been smoothed out as noise in the input, as shown in Figure 8 (left). The structure brush is directly applied on the noise layer \(I_n\) since the structure residual is hard to identify on the latent image \(I_l\), but is obvious for the user to see on \(I_n\). We believe this is the first interactive approach where the user edits the noise layer for removing denoising artifacts in the output latent image.

For simplicity of notations, we still use \(I\) to denote various regions where the user brush is applied. Let the user-selected region using the structure brush be \(I_s\). Our filter is applied on \(I_s\) (with the help of \(I_l\)) to decompose it into a structure residual \(I_{s'}\) and a noise signal \(I_{n'}\). The image-noise separation result is thus updated as

\[
I_{l'} = I_{l} + I_{n'}, \quad I_{s'} = I_{n'}.
\] (9)

The values of the three parameters \(\sigma_c\), \(\sigma_d\) and \(\sigma_s\) of the joint image-noise filter can be manually modified by the user for each brush, or automatically evolve based on a gesture control mechanism. The default values of the three parameters are set conservatively \((\sigma_c = 25, \sigma_d = \sigma_s = 7\) in our system), thus strong structure signal in \(I_s\) can be extracted using only one brush stroke. For weaker structure signal, all three parameters need to be decreased accordingly. This can be achieved by having the user brush over the same region multiple times like applying a real rubber eraser. The system will then progressively and linearly decrease the values of the parameters, based on how many times the user has brushed over the same region. As a result, the structure residual in \(I_s\) will be gradually reduced until a random noise region \(I_{n'}\) is produced. In case \(I_s\) is over-smoothed, the user can use the undo operation to get back the previous result.

4.3 The Smooth-Region Brush

The purpose of the smooth-region brush is to remove visible noise residual \(I_{n'}\) from the smooth regions of the latent image \(I_l\). It works in principle similar to the structural brush, except that it is applied on the latent image \(I_l\) rather than the noise image \(I_n\), as shown in Figure 8 (right). The user selects a region \(I_l\) where noise residual is present, on which the filter is applied. The image-noise separation
result is updated as

\[ I^*_i = I_i - I^0_i, \quad I^*_n = I_n + I^0_n. \]

(10)

Similar to the structure brush, parameters of the filter in the smooth-region brush can also be adjusted automatically using the same gesture control mechanism.

5 Evaluations

5.1 Quantitative Evaluation

Figure 10: The test dataset. The close-ups of the regions highlighted in red are shown in Figure 12.

We perform quantitative evaluations on our system. A dataset containing 17 images chosen from the Berkeley segmentation dataset [Martin et al. 2001] was constructed, as shown in Figure 10. For each test image, additive white Gaussian noise (AWGN) with \( \sigma = 5\% \) is added. We first apply five widely-used denoising packages, including the BLS-GSM method [Portilla et al. 2003], Photoshop [Adobe Systems 2008b], Noiseware [Imagenomic Inc. 2008], Neat Image [ABSoft Inc. 2008] and Noise Ninja [PictureCode Inc. 2008], to generate the initial image-noise separation results. We then apply the proposed interactive system to refine the initial image-noise separation. Five subjects are chosen for this task, all of them are familiar with brush-based user interface for image editing tasks. We give each subject a brief tutorial on our system, and allow the participant to test the UI freely until he/she feels comfortable with performing the task. We then ask the participant to work on each example generated by a specific initial denoising algorithm. The time allowed for user interaction is limited to 5 minutes for each example.

Figure 11 shows the PSNR gains our system achieves for each initialization method. The average PSNR gain over the entire dataset is 2.61dB, which suggests that our system consistently and significantly improves the image-noise separation results produced by various existing methods. In this figure we also compare our results with those reported in the recently proposed region-based approach [Liu et al. 2008] (average PSNR difference is 1.42dB), which suggests that our system can generate results of significantly higher quality that the state-of-the-art automatic approaches cannot achieve. In Figure 12, we show four close-up comparisons of local image patches generated by different algorithms, which indicate that our improvements are visually quite noticeable. We further quantize the differences using the recently proposed Structural Similarity Index (SSIM) [Wang et al. 2004], which shows similar performance improvements. Please refer to the supplementary slides for the numerical results using SSIM.

To demonstrate that the proposed joint image-noise filter is effective for both flat image regions as well as image structures and textures, we categorize all pixels in the 17 test images into 10 groups based on their gradient magnitudes, and compute the average PSNR gain in each group. The result is shown in Figure 13, which clearly suggests that the proposed system achieves consistent improvement across all gradient scales. In many cases larger gains are achieved for high gradient levels, indicating that previous denoising algorithms tend to destroy high frequency contents in order to generate a clean image, while our system can effectively restore them.

\[ \text{PSNR difference} \geq 0.5\text{dB} \] is generally considered perceptually noticeable. For instance the MPEG committee used a threshold of 0.5dB PSNR to decide it is worth to incorporate a given coding optimization [Salomon 2005].
Figure 11: Quantitative evaluation of different algorithms computed on the dataset, using PSNR index. For each test the short red bar on the top indicates the performance gain after applying our system. For [Liu et al. 2008] the short blue bar on the top indicates the difference between the highest PSNR generated by our system and the PSNR reported in [Liu et al. 2008].

Figure 12: Close-up comparisons of results generated by various methods. Our results faithfully preserve high-frequency image contents in each example. Please refer to the high resolution electronic version of the paper for a clear comparison.

Figure 13: Average PSNR gain produced by our system for pixels in different gradient level groups.

5.2 Experiments with Real Noise

To demonstrate the effectiveness of our system on real noise, in Figure 14 we apply the system to the non-flash noisy image used in [Petschnigg et al. 2004], where a flash and non-flash image pair is used to generate a denoising result shown in Figure 14(b). This is a challenging example, as the denoised image in [Petschnigg et al. 2004] still exhibits a noticeable amount of noise. We first use Noiseware to generate an initial denoising result shown in Figure 14(c), and then apply our system to generate a clean image shown in Figure 14(d). Note that our system only uses a single noisy image as input, but the denoising result has a higher quality than the one generated by the original image pair approach.

Figure 15 shows an example where the input image is captured by a DSLR at a high ISO setting. We preprocess the input image using Noiseware\footnote{The intermediate result is shown in the supplementary slides.}, and then use our system to generate the final result shown in Figure 15(b). Comparing the noise layer before and after applying our system (Figure 15(c) and (d)), it is clear that our system is efficient at recovering detailed image structures.

6 Noise-Consistent Image Editing

For a noisy input image, a variety of image editing tasks can benefit from a high quality image-noise separation. In Section 5 we have demonstrated that our system can achieve high quality image denoising as an immediate application. In this section we show that
of CCD cameras [Laroche and Prescott 1994]. These patches are color filter pattern sensing and demosaicing in the imaging process in a small neighborhood to have spatial correlations due to CCD ent. Thus, it is inadvisable to directly add the noise layer I′ to the original latent image I, (See Equation 2) and the original latent image I′ may be different. To increase the randomness, a random sampling is applied to choose one noise patch out of top 10 candidates.

In many patch-based texture or image synthesis work [Efros and Freeman 2001], a compatibility measure is usually defined to measure how naturally two patches can be stitched together without introducing a seam between them. However in our system, we found such a measure is not necessary, since the noise map I′ generated by our system contains no large-scale structures, thus it is very unlikely to generate a noticeable seam by stitching together two noise patches. By only employing the patch distance measure described above, the noise synthesis is simple, fast, and generates visually acceptable results, as we will demonstrate in the following applications.

6.2 Applications

Sharpening and Blurring

Directly sharpening or blurring a noisy image will boost or reduce the noise level and lead to undesirable results, as shown in Figure 16. This problem can be alleviated by performing image-noise separation first and then only applying the operation on the latent image. However, artifacts will still present in the resulting image if the image-noise separation is not sufficiently good (Figure 16, top). Using our system, a high quality image-noise separation can be achieved, thus the noisy image can be sharpened or blurred without affecting the noise level.

Image Retargetting

Noise characteristics can be easily affected by resizing the input image, such as creating a thumbnail. However, in many applications, preserving the noise characteristics in the thumbnail is desirable, since the user can quickly preview the noise level without loading the original version. Our system can be used to achieve this goal. First, a high quality image-noise separation is obtained using our system. Then, our noise synthesis can be applied to re-generate a noise layer for the thumbnail image, as shown in Figure 17 (top).

Noise Transfer

Transferring noise from one image to another requires accurate image-noise separation on the source image, and thus can benefit from using our system, as shown in Figure 17 (bottom). Although the grain matching command in Adobe After Effects can be employed for this purpose, the parametric grain model cannot realistically capture the unique characteristics of the noise present in the source image. In contrast, by applying high quality image-noise separation followed by nonparametric noise synthesis, our system can achieve consistent noise transfer.

7 Limitations and Discussions

Given its simplicity and effectiveness, our system has a great potential to be incorporated in existing systems for a variety of applications. However, the current system does present some limitations that one should consider when applying it in conjunction with other technologies.
Choice of the initialization method. Our system takes as input an initial image-noise separation result, thus can follow up a wide range of denoising packages and achieve substantial improvements, as we have shown in this paper. On the negative side, the quality of the final result will be affected by the selection of the initialization method. From figure 11, it is clear that the final PSNRs generated by our system vary based on the quality of initial denoising. In general higher quality initial denoising generally leads to higher quality output in our system, and vice versa. Furthermore, using a low quality denoising input will require much more user input. In an extreme case, directly applying the system to an input noisy image without a reasonable initial denoising will require tremendous user effort, and may not lead to a desired result.

In practice, we suggest that the initial denoising method should be chosen not only by its PSNR performance, but also by its speed, memory consumption, whether or not being optimized for a specific type of noise, etc. Our system serves as a general tool and will support various packages one may choose.

Unstructured image details. Our system assumes that high frequency image signal is structured. This may not always be true, and the system will push unstructured high frequency details into the noise layer. For instance, the details of a sand beach image may be misclassified as noise by our system and be over-smoothed. To avoid this the user may need to apply a protection mask over this type of image regions.

8 Conclusion and Future Work

We propose joint image-noise filtering for image-noise separation. Our method jointly optimizes the image and the noise layer, thus is in sharp contrast with existing denoising techniques which focus exclusively on optimizing the latent image. The proposed joint filter is integrated into an easy-to-use interface where artifacts on both layers can be easily removed using mouse brushes. Furthermore, by deploying our system in noise-consistent image editing, we demonstrate high-quality image editing results where noise can be faithfully generated, preserved, or transferred.

In the future, we plan to investigate how to combine the proposed filter with a specific initial denoising algorithm for applications where fully automatic high quality denoising results are needed.
9 Acknowledgements

We thank the anonymous reviewers for helping improve this paper. The work was performed when Jia Chen was interning at Adobe’s Advanced Technology Labs. This work is partially supported by the Research Grant Council of Hong Kong Special Administration Region, China: HKUST620309.

References

WEISS, Y., AND FREEMAN, B. 2007. What makes a good model of natural images. In CVPR.