Interactive Local Adjustment of Tonal Values

Dani Lischinski
The Hebrew University

Zeev Farbman
The Hebrew University

Matt Uyttendaele
Microsoft Research

Richard Szeliski
Microsoft Research

Figure 1: Three different interpretations generated from the same digital negative using our tool. Left: warm sky, high exposure in the foreground. Middle: cooler sky, medium exposure in the foreground. Right: an even cooler sky, very little exposure in the foreground leaving almost no detail but the silhouette. RAW image courtesy of Norman Koren, `www.normankoren.com`.

Abstract

This paper presents a new interactive tool for making local adjustments of tonal values and other visual parameters in an image. Rather than carefully selecting regions or hand-painting layer masks, the user quickly indicates regions of interest by drawing a few simple brush strokes and then uses sliders to adjust the brightness, contrast, and other parameters in these regions. The effects of the user's sparse set of constraints are interpolated to the entire image using an edge-preserving energy minimization method designed to prevent the propagation of tonal adjustments to regions of significantly different luminance. The resulting system is suitable for adjusting ordinary and high dynamic range images, and provides the user with much more creative control than existing tone mapping algorithms. Our tool is also able to produce a tone mapping automatically, which may serve as a basis for further local adjustments, if so desired. The constraint propagation approach developed in this paper is a general one, and may also be used to interactively control a variety of other adjustments commonly performed in the digital darkroom.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.3.9 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.4.3 [Image Processing and Computer Vision]: Enhancement; I.4.9 [Image Processing and Computer Vision]: Applications;

Keywords: digital darkroom, high dynamic range imaging, image editing, stroke-based interface, tonal adjustment, tone mapping

1 Introduction

From the moment the camera shutter is released until the resulting photograph is printed or displayed on a screen, an image undergoes a variety of tonal and color adjustments (e.g., brightness, contrast, saturation, and white balance). Both the digital camera and the photo lab machines use sophisticated image analysis and processing algorithms to make sure that the grass looks green, the sky looks blue, and the flesh tones look just right. In addition to correction and enhancement, the ability to adjust tonal values provides photographers with an invaluable means for creative expression and for perfecting their photographs.

Consider, for example, the three possible renditions of the backlit scene shown in Figure 1. Each of these three images conveys a different mood and tells a different story. Yet all three images were "developed" from the same *digital negative* by applying different local adjustments of exposure, saturation, and color temperature.

In the wet (chemical) darkroom of old, dodging and burning techniques were the only available tools for achieving this kind of local control [Adams 1983]. With the advent of the digital darkroom (image editing software such as Adobe Photoshop [Adobe Systems, Inc. 2005]), photographers now have a large arsenal of adjustment tools at their disposal, offering incredible power and many additional ways for manipulating their photographs. Unfortunately, mastering these tools involves a steep learning curve, and the digital photography workflow is tedious and time consuming even for the professionals, as evidenced by the large number of classes, textbooks, and detailed tutorials on the world wide web.

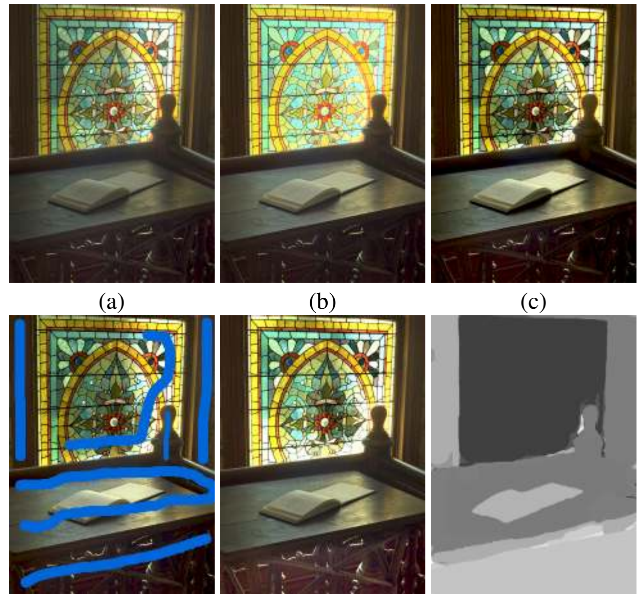
In this paper, we show how a simple *stroke-based* interface may be used to rapidly and intuitively perform local adjustment of the tonal values (and other visual parameters) in an image. Our goal is to provide digital photographers with a tool that is easy to use yet powerful, and that does not inhibit their creativity or undermine their ability to control different aspects of the results. Our work is formulated in an *image-guided energy minimization* framework, which aims to propagate a small number of brush strokes (along with associated parameter values) to a set of complete adjustment maps defined over the entire image.

Similar stroke-based interfaces have recently been proposed for foreground extraction [Li et al. 2004; Rother et al. 2004], as well as for alpha matting [Wang and Cohen 2005]. However, challenging images may contain many regions, each requiring a different adjustment. Explicitly extracting each individual region is an overkill, and there is also the question of how to properly blend the manipulated regions back together. Our work is more closely inspired by the Digital Photomontage [Agarwala et al. 2004] system, and by the colorization work of Levin *et al.* [Levin et al. 2004].

Our techniques were primarily designed to handle high dynamic range (HDR) images, such as those constructed from multiple exposures [Reinhard et al. 2005]. However, they are fully applicable to ordinary, single exposure photographs. Most professional and advanced amateur photographers nowadays prefer to shoot in "camera raw" (RAW) mode, and then use RAW conversion software to tone map and color-correct their images. Our tool fits well into this workflow. We also demonstrate that it can be used for quickly fixing up casual snapshots. As sensor technology continues to evolve, digital cameras will capture increasingly higher dynamic range. Thus, tomorrow's "digital negatives" will present even more need (and opportunity) for high-quality tonal adjustment.

Of course, a large number of sophisticated automated tone mapping algorithms have already been developed in the computer graphics community (see the Related Work section). However, these algorithms do not provide any direct or local control over the result. It is our belief that photographers will always want the option to creatively manipulate and perfect their photographs, and that the tools and techniques we develop in this paper make this process much easier and, as a result, far more effective.

In summary, our specific contributions include: (i) a new interactive approach to tonal adjustment of images, designed with the creative workflow of digital photographers in mind; (ii) a simple, unified, and intuitive user interface, which enables local and direct manipulation; (iii) a fast iterative solver and a method for precomputing stroke influence functions that enables tonal adjustment at interactive rates; (iv) an automatic tone mapping mode that compares favorably with previous tone mapping operators; (v) a framework applicable to other kinds of local image manipulations beyond tonal adjustment.



(a) (b) (c)

(d) (e) (f)

Figure 2: (a) Tone mapping produced by the gradient domain method of Fattal *et al.* [2002]. (b) Tone mapping produced by Reinhard's operator [2002]. (c) Tone mapping produced with our interactive tool. We refer the reader to the electronic version of the paper in order to better appreciate the differences between the images in this and subsequent figures. (d) The strokes used to generate the result in (c). (f) Hand-painted Photoshop layer mask and the corresponding exposure blend (e). (HDR image © Industrial Light & Magic. All rights reserved.)

2 Related work

2.1 Tone mapping operators

Many different tone mapping (or tone reproduction) operators have been proposed in the computer graphics and image processing literature over the years. We refer the reader to [Reinhard et al. 2005] for a survey of these methods. The different approaches may be roughly classified into global operators, which essentially utilize a curve to map each pixel to a display value, [Ward Larson et al. 1997; Reinhard et al. 2002; Drago et al. 2003], and local (spatially variant) operators, [Pattanaik et al. 1998; Tumblin and Turk 1999; Ashikhmin 2002; Durand and Dorsey 2002; Fattal et al. 2002; Reinhard et al. 2002; Li et al. 2005].

Most of the recent operators are capable of effectively mapping HDR radiance maps into a displayable low dynamic range image. Global operators are usually faster, but spatially variant operators are better at preserving local contrasts across the entire dynamic range. However, local operators sometimes introduce visual artifacts into the tone mapped result, and tend to visualize all of the detail in the scene, which may not be what the photographer intended. Although these tone mapping operators are typically regarded as automatic, many of them require some parameter tweaking to obtain the best result for a particular image.

All of the operators we are aware of suffer from lack of direct local control, as they operate on the entire image at once. Although the user has a certain degree of control over the outcome via the provided parameters, there is no way for her to directly specify the desired brightness or contrast of a particular region in the image, or to confine the manipulation of brightness just to that region. Thus, the tone mapping process typically involves several trial-and-error iterations. In each iteration the entire image is subject to change, and therefore the process is not guaranteed to converge to a subjectively satisfactory result.

Consider the images in Figure 2 as an example. Images 2a and 2b were produced from an HDR radiance map using two representative tone mapping operators. We experimented with the available control parameters to obtain the result that looked best to us. Both operators succeed in compressing the dynamic range to the point were detail is visible in all image regions. However, we believe that the image in Figure 2c (produced using our interactive system) is a much better photographic reproduction of the scene: the texture of the stained glass is much richer, and the contrast on the desktop is more pronounced. The feeling that a considerable amount of light is reflected from the desktop and the open notebook is effectively conveyed. Our interactive tool allowed the artist to manipulate the different regions, such as the window and the desktop, in a direct manner until the desired result was achieved. Whether or not our result is indeed a better photograph is a matter of taste. Our point is that the image on the right is what the artist *wanted*, and that it was not possible to obtain this result with any of the other tone mappers, no matter how much time was spent tweaking their control parameters.

Finally, we note that while several authors have previously discussed "interactive tone mapping" [Durand and Dorsey 2000; Artusi et al. 2003; Goodnight et al. 2003], the goal of these papers was to achieve interactive performance with tone mapping operators, rather than providing the user with intuitive, direct, and local control, which is the goal of our work.

2.2 Interactive image manipulation tools

Digital photographers commonly utilize interactive image manipulation and editing packages, such as Adobe Photoshop, in their workflow [Reichmann 2005]. Such packages feature an impressive array of interactive image adjustment controls, such as “levels”, “curves”, brightness and contrast sliders, color balance controls, and many others. Images are manipulated directly and immediate visual feedback is provided. In order to confine the effect of a manipulation to a particular region in the image, the user must first specify the affected region. This may be done using any of the available selection tools, and converting the selection into a *layer mask*. Layer masks may also be painted directly using the available painting tools. By stacking a number of adjustment layers, each with its own layer mask, the user is able to achieve complete local and direct control over the outcome.

Image editing packages also offer the dodge and burn brushes, which may be used to increase or decrease the exposure locally. Again, these brushes must be used with great care and precision (or applied to already masked regions).

Unfortunately, creating masks by selecting or by painting is a tedious and time-consuming task, requiring considerable skill and precision. To avoid visual artifacts, the boundary of the mask must follow the boundary of the region very closely, making it quite difficult to construct masks for regions whose boundary is complex or fuzzy. Furthermore, the mask boundary must be carefully feathered to produce a seamless and artifact-free blended result. As a concrete example, consider again the desk scene in Figure 2. When an experienced Photoshop user was asked to match the image shown in 2c, he began by generating two different exposures in two separate layers and then hand-painted the layer mask for blending them together. The mask is shown in Figure 2f, next to the resulting image. The entire process took about 15 minutes.

Compared to the interactive tools described above, our tool offers the user the same degree of direct and local control, but without the need to make precise selections or explicitly paint masks. Instead, our user interface is stroke-based, Thus, the user is presented with a much simpler, unified, and intuitive interface: the desired adjustment parameters are specified directly in a small number of locations in the image, and a spatially varying adjustment map for the entire image is computed in a highly interactive manner. For example, it took one of the authors less than 3 minutes to generate the image in Figure 2c, using a small number of brush strokes shown in Figure 2d.

The nature of the interaction in our tool resembles that of interactive digital photomontage [Agarwala et al. 2004], where the user uses strokes to indicate which parts of a set of photographs should be combined into a single composite result. Graph-cut optimization is then to find good seams between different regions, and gradient domain fusion is used to blend over those seams. In contrast, in our approach, the pixels in the final result come from a continuous space of all possible adjustments of the input image, rather than from a discrete set of images. Furthermore, our approach requires neither explicitly finding seams, nor blending across them. Instead, we use a more unified approach, where constraints set by the user’s strokes are propagated to the entire image in a piecewise smooth manner.

Our work is most closely related to the colorization method of [Levin et al. 2004], which also uses scribbles to define constraints and propagates them to the entire image by solving an optimization problem. However, we derive a different (sparser and better motivated) affinity function for the propagation, and propose carefully designed numerical methods suitable for interactive performance. Furthermore, our work generalizes the idea of propagating chroma channels to propagation of general adjustment parameters.

3 The interactive tone mapping tool

The goal of our tool is to enable the photographer to easily adjust exposure, along with several other common types of adjustments, in a spatially varying manner. The purpose of these adjustments might be to reveal detail that would otherwise be lost due to over- and under-exposure, to enhance the contrast, correct the saturation of the colors, or to achieve certain artistic effects. Our proposed workflow is summarized below:

1. Load a digital negative, which could be a camera RAW file, an HDR radiance map, or an ordinary image.
2. Indicate regions in the image that require adjusting.
3. Experiment with the available adjustment parameters until a satisfactory result is obtained in the desired regions.
4. Iterate steps 2 and 3 until satisfied with the entire resulting image.

Steps 2 and 3 may be carried out in any order. This workflow is illustrated by the sequence of images in Figure 3, as well as in the supplemental video.

Several requirements must be met for the above workflow to be effective. First, the available set of adjustments must have sufficient expressive power. Second, the mechanism for indicating the regions being adjusted should be simple, intuitive, and easy to use. Third, the user must be provided with quickly generated previews of the overall result. Finally, it should be easy for the user to modify (or undo altogether) previously specified adjustments.

In our current implementation, the available adjustments are: exposure (applying a multiplicative factor to the radiance values of a pixel), contrast, saturation and color temperature. In principle, our tool is not limited to these adjustments and could easily be extended to accommodate a variety of other common operations, such as blurring and sharpening.

In our tool, the user indicates affected regions by simply scribbling on the image (see Figure 3). Several different types of brushes and their semantics are described in Section 3.1. Each brush stroke along with the specified adjustment parameters is interpreted as a *soft constraint* on the desired outcome. The adjustment parameters are then propagated to the entire image by solving an image-guided minimization problem, as described in Section 3.2. Previews are rapidly generated by computing an approximate solution, as described in Section 3.3.

3.1 Strokes and brushes (region selection)

Our prototype implementation supports several different types of brushes that may be used to set constraints.

Basic brush: only pixels directly covered by the brush stroke are constrained. We currently use a flat brush profile, assigning a weight of 1.0 to each covered pixel, although arbitrary brush profiles could easily be incorporated.

Luminance brush: this brush is used to apply a constraint to pixels whose luminance is similar to those covered by the brush stroke. Let μ be the mean lightness (CIE L^*) value of the pixels under the stroke, and let σ be some threshold (set by the user via the mouse scroll-wheel). A pixel with a lightness value of ℓ is selected only if $|\mu - \ell| < \sigma$, and the weight of the constraint is set to 1.0. For smoother results a Gaussian falloff may be used instead:

$$w(\ell) = \exp(-|\ell - \mu|^2/\sigma^2).$$

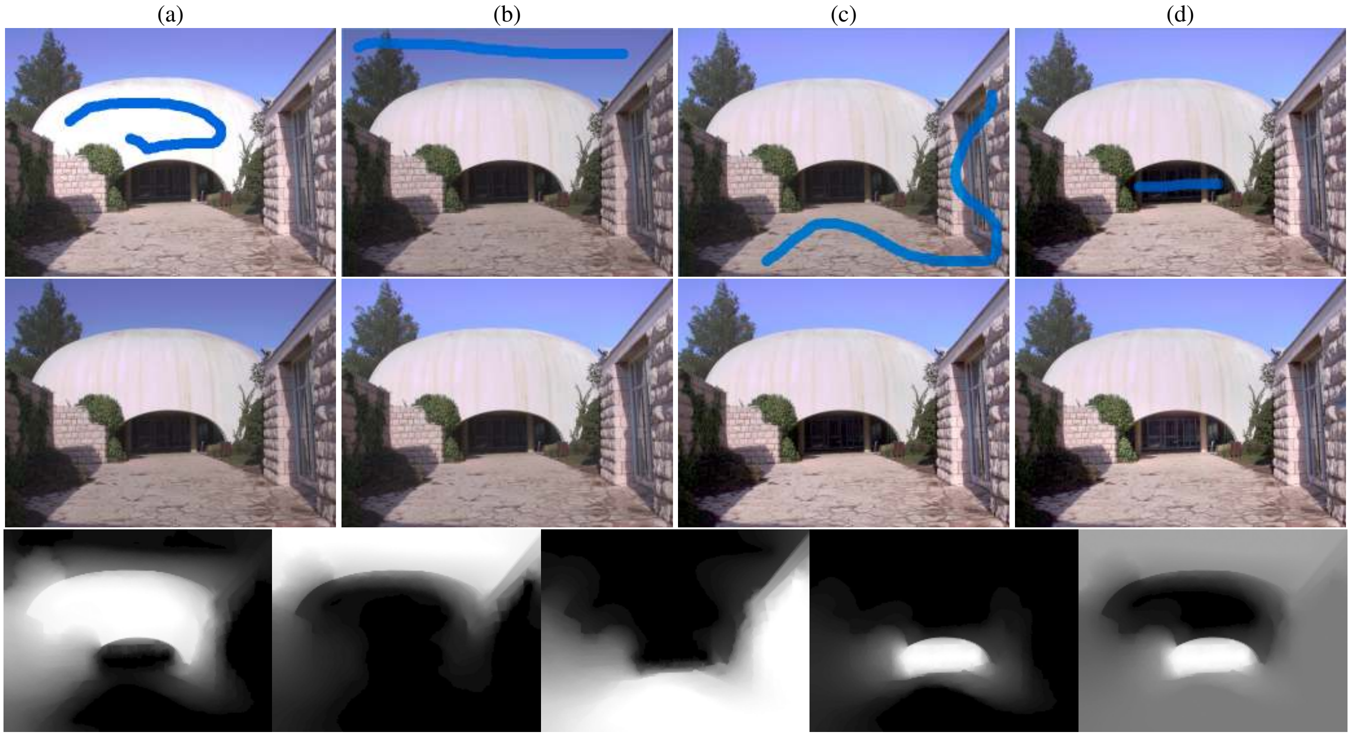


Figure 3: Step-by-step demonstration of our workflow. Top row: strokes drawn by the user; second row: the result after adjusting one or more parameters for the corresponding stroke. We refer the reader to the electronic version of the paper in order to better appreciate the differences between these images. (a) The user draws a stroke across the overexposed dome, and decreases the exposure to make some texture visible. Since this is the first and only constraint so far, the entire image is darkened. (b) The user draws a stroke across the sky and increases the exposure. The dome appears unaffected. (c) The next stroke is drawn across the foreground, followed by a slight adjustment in the exposure and an increase in the contrast. (d) Finally, the exposure and contrast are increased under the entrance arch. At any stage the user may select one of the previous strokes and modify the parameter values associated with the stroke, or delete the stroke altogether. The bottom row shows the four influence functions corresponding to the strokes in (a)–(d), and the rightmost image is the resulting exposure adjustment map. Note how strong edges in the image limit the influence of the individual strokes, and correspond to discontinuities in the final adjustment map.

Lumachrome brush: this brush is similar to the luminance brush, but it accounts for similarity in the chromaticity channels as well. The computation is performed in the CIE $L^*a^*b^*$ color space. This brush is demonstrated in Figure 4.

Over-exposure brush: We also experimented with a brush that selects all over-exposed pixels inside a given region of the image. The user draws a stroke surrounding the region of interest, and all of the pixels inside this region whose currently mapped value exceeds a certain threshold are selected. Under-exposed pixels may also be selected in the same manner.

Finally, the tool also offers a new automatic tone mapping algorithm, described in Section 4. The results produced by this algorithm may serve as a good starting point for further interactive local adjustments.

3.2 Constraint propagation

For clarity of presentation, let us assume for the moment that our goal is to compute only the spatially varying exposure function f, which is a scalar function that specifies (in terms of f-stops) how the exposure of each pixel is to be adjusted. Our first requirement is that f comes close to satisfying the user-specified constraints, making it a scattered data approximation problem. At the same time, we would like the function to be as smooth as possible, since unnecessary oscillations in the exposure will result in blotchy tone mapped images. Rapid changes in the exposure should be allowed, however, across significant edges in the image, with the rate of change in f being commensurate with the rate of change in image brightness at each pixel. We attempt to achieve these objectives by minimizing the following quadratic functional:

$$f = \arg\min_f \left\{ \sum_{\mathbf{x}} w(\mathbf{x}) \left(f(\mathbf{x}) - g(\mathbf{x})\right)^2 + \lambda \sum_{\mathbf{x}} h\left(\nabla f, \nabla L\right) \right\} \quad (1)$$

The first term of this functional is the *data term*, responsible for satisfying the user-specified non-overlapping constraints. The weight function $w(\mathbf{x})$ indicates which pixels are constrained, and specifies a weight (between 0 and 1) for each constrained pixel. The target exposure values at the constrained pixels are given by $g(\mathbf{x})$.

The second term is the *smoothing term*, whose objective is to keep the gradients of the exposure function f as small as possible, except where the underlying image has significant gradients in its log-luminance channel L. We use the log-luminance to make the results of interpolating a set of strokes independent of the overall brightness of the scene, which may depend on potentially unknown exposure parameters or be set to an arbitrary value for an HDR image. The smoothing term we use is

$$h\left(\nabla f, \nabla L\right) = \frac{|f_x|^2}{|L_x|^\alpha + \varepsilon} + \frac{|f_y|^2}{|L_y|^\alpha + \varepsilon} \quad (2)$$

Here the subscripts x and y denote spatial differentiation of the functions f and L. The exponent α controls the sensitivity of the term to

Figure 4: When the region to be manipulated is fragmented but consists of pixels with like colors, it is convenient to use our Lumachrome brush. The green scribble shows the user-drawn stroke, and the constrained pixels are shown in magenta. (HDR image © Industrial Light & Magic. All rights reserved.)

the derivatives of the log-luminance image, and ε is a small constant to avoid division by zero. The relative weight of the two terms in the functional is controlled by the parameter λ. The effect of these parameters is demonstrated in Figure 5. In our current implementation we use the default values $\lambda = 0.2$, $\alpha = 1$, and $\varepsilon = 0.0001$, which were used to produce all of our results (except where noted otherwise).

The bottom right image in Figure 3 shows the exposure adjustment map at the end of the interactive process. Note that the adjustment map is indeed piecewise smooth, with discontinuities corresponding to significant edges in the image.

Using standard finite differences for the spatial derivatives of f results in a quadratic expression in f, whose unique minimum is obtained by solving the linear system

$$\mathbf{A}f = b, \tag{3}$$

where
$$\mathbf{A}_{ij} = \begin{cases} -\lambda \left(\left| L_i - L_j \right|^{\alpha} + \varepsilon \right)^{-1} & j \in N_4(i) \\ w_i - \sum_{k \in N_4(i)} \mathbf{A}_{ik} & i = j \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

and
$$b_i = w_i\, g_i. \tag{5}$$

Here, the subscripts i and j denote pixels i and j in the image, and $N_4(i)$ are the 4-neighbors of pixel i.

3.3 Fast approximate solution

The sparse matrix $\mathbf{A}$ is symmetric positive definite. Thus, system (3) may be solved iteratively by preconditioned conjugate gradients (PCG) [Saad 2003]. The convergence of PCG strongly depends on a good choice of preconditioner. An incomplete Cholesky factorization of $\mathbf{A}$ is often used for this purpose. However, due to our requirement for rapid feedback we cannot afford to recompute the preconditioner every time the user adds or removes a scribble, thereby changing $\mathbf{A}$. Fortunately, it turns out that a good preconditioner may be computed ahead of time. Note that the matrix $\mathbf{A}$ may be written as a sum $\mathbf{A} = \mathbf{L} + \mathbf{W}$, where $\mathbf{W} = \mathrm{diag}(w_1, \ldots, w_n)$ is a diagonal matrix containing the weight of the constraint at each pixel. The matrix $\mathbf{L}$ depends only on the original image, and only $\mathbf{W}$ is affected by the specific constraints. Thus, we compute only once the incomplete Cholesky factorization of the fixed matrix $\mathbf{L} + \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, rather than that of the changing matrix $\mathbf{A}$. The resulting factor matrix is then used as the preconditioner in all subsequent invocations of PCG. We found that this preconditioner yields almost the same convergence rate as the incomplete Cholesky factor of $\mathbf{A}$.

To further speed the solution, our solver utilizes a multigrid-like approach. We compute a series of matrices $\mathbf{A}^{(0)}, \ldots, \mathbf{A}^{(k)}$ corresponding to progressively coarser versions of the image. After each change to the constraints, we begin by directly solving the lowest resolution version of the system. The resulting solution is then upsampled and passed to the next level. At each finer level the solution is quickly updated by performing a small number of PCG iterations, and forwarded further, until the current preview resolution is reached. In our current implementation the typical preview resolution is around 600×400, and it takes a fraction of a second to obtain the first usable preview once a constraint has been added or modified by the user.

All of the results in this paper were produced using the coarse-to-fine approach described above. However, we have also experimented with locally adapted hierarchical basis function preconditioners [Szeliski 2006] as an alternative means to perform the solution, and found them also to be well-suited for the task.

Basis function decomposition. When several tonal parameters are being interpolated based on the user-specified constraints, the iterative solver must be invoked separately for each parameter. An alternative approach is to precompute the *basis* or *influence function* associated with each stroke (or layer, if strokes are grouped into layers), and then compute the resulting parameter maps as linear combinations of these basis functions.

To explain why this approach works, note that the weights $w(\mathbf{x})$ in (1) can be written as the sum of per-constraint weights $w_l(\mathbf{x})$,

$$w(\mathbf{x}) = \sum_l w_l(\mathbf{x}). \tag{6}$$

We can thus re-write the right hand side of the linear system (3) as

$$b = \sum_l w_l\, g_l, \tag{7}$$

where w_l is the vector of per-pixel weights of layer l, and g_l is the corresponding (constant) target value.

We can pre-compute the influence function u_l for layer l by using the weights as the right-hand side ($b = w_l$) and solving the linear system (3). Several influence functions are shown in Figure 3. Once this has been done, the parameter map f_k resulting from a set of constraints g_{kl} (where k is the k-th parameter) may be computed as a linear combination of the influence functions:

$$f_k = \sum_l g_{kl} u_l. \tag{8}$$

While this approach may be more computationally efficient when the number of strokes/layers is smaller than the number of different adjustment parameters, it becomes particularly effective once the user stops scribbling and starts adjusting the sliders corresponding to the individual parameters. The new mapped image can now be recomputed without re-invoking the iterative solver, as is demonstrated in our accompanying video. In fact, it is not even necessary to compute the basis functions for all of the constraints. If instead, only the basis for the constraint whose parameters are being adjusted, u_l, is precomputed, changing the value of the k-th slider, $g'_{kl} \leftarrow g_{kl} + \Delta g_{kl}$, results in a new solution

$$f'_k = \mathbf{A}^{-1} b'_k = \mathbf{A}^{-1}(b_k + \Delta b_k) = f_k + \mathbf{A}^{-1} \Delta g_{kl} w_l = f_k + \Delta g_{kl} u_l.$$

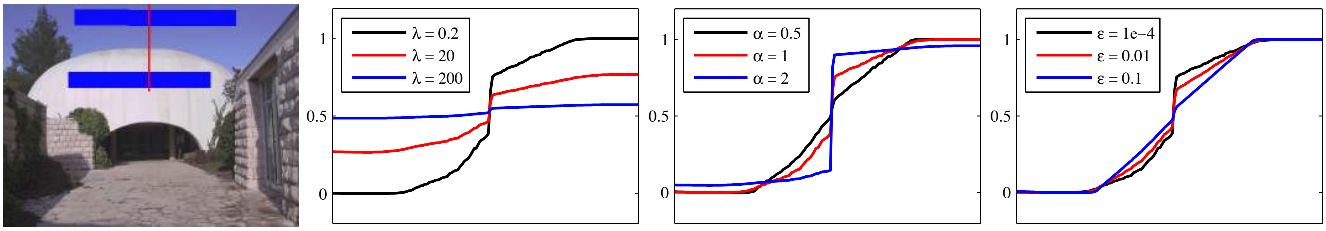


Figure 5: The effect of the parameters on the interpolation. In the image on the left, the top stroke constrains the exposure to 1, while the bottom stroke constrains it to 0. The resulting exposure map is plotted along the red vertical line for different values of λ, α, and ε.

4 Automated initialization

When working on an HDR image, it is useful to present the user with some initial automatically computed tone mapping, and then allow the user to adjust this mapping to her liking. We therefore extended our basic tool with an automatic algorithm for setting up the constraints in an image. Our idea is inspired by Ansel Adams' Zone System [Adams 1981; Adams 1983]. Reinhard *et al.* [2002] also use the basic conceptual framework of this system to manage choices in their photographic operator. We follow the Zone System even more closely, by explicitly specifying a target exposure for each individual zone in the negative. Once the initial mapping has been computed, the user is free to select any of the zones and adjust its exposure, as well as other parameters. Alternatively, she can accept the result and then apply further adjustments via the stroke-based mechanism described earlier.

Specifically, we decompose an image into a collection of zones, $1,\ldots,Z$, determine an appropriate target exposure value e_i for each zone, and associate each pixel in the zone with this target value. Simply applying the target exposure to each pixel would result in visible discontinuities across boundaries between different zones. A smoother exposure field is obtained by providing the target values of the pixels as soft constraints for our solver, with the weights w_i in (5) set to a small value (typically 0.07).

The number of zones Z in a digital negative is determined simply by the range of the log-luminance values of the image:

$$Z = \lceil \log_2(L_{\max}) - \log_2(L_{\min} + \varepsilon) \rceil .$$

Next, a representative luminance value R_z is chosen for each zone z, and a monotonic non-linear curve f is applied to map R_z to a target value $f(R_z)$. The target exposure value (in f-stops) is then simply $\log_2(f(R_z)/R_z)$.

There are many possible options for choosing a representative value for each zone. We found that good results are obtained by taking the median luminance of the pixels in the zone. We also found that a wide variety of different non-linear monotonic curves may be used for f. We experimented with Reinhard's global operator [2002], a simple log function, and a variety of power curves. Figure 6 shows some of the results we obtained using Reinhard's operator to compute the target exposures. It is interesting to compare these results with images obtained by directly applying Reinhard's operator to each pixel. The results are very similar in the darker parts of the image, but contrast in the brighter regions is preserved better with our method. This is a consequence of applying a nearly constant exposure for all the pixels in a zone, rather than gradually varying the exposure from one pixel to the next.

Figure 6: Our automatic mode results (left), compared to Reinhard's global operator (right). The overall look is similar, but note the improved contrast in the brighter areas. (Doll HDR image courtesy of Yuanzhen Li.)

5 Results

Several of the results generated with our tool were already shown earlier in the paper[1]. To produce the images in Figure 1, we used the basic brush to adjust different regions of the sky, and the lumachrome brush to select the foreground rocks and tree, as well as the bright parts of the sky behind the tree. Experimenting with exposure, saturation, and color temperature yielded the three renditions of this scene.

Figure 2c shows a tone mapping of an HDR image, produced interactively with our tool, using only the simple brush strokes shown in Figure 2d, adjusting exposure and contrast. A comparison with previous automatic tone mappers indicates that our tool provides the kind of control needed to produce more compelling photographic renditions from HDR images. Additional comparisons of this kind are included in the supplemental materials and the companion video.

Figure 3 demonstrates that our constraint propagation mechanism indeed provides the ability to control local regions in the image without resorting to carefully created masks. Adjusting the parameters associated with a particular brush stroke affects the region

[1] The reader is referred to the ACM SIGGRAPH 2006 Full Conference DVD-ROM, which includes full-resolution versions of the images in this paper, additional examples, as well as stroke and zone visualizations.

Figure 7: These images were adjusted by using our automatic tone mapper to produce the initial mapping, followed by interactive adjustments by the user. (HDR images courtesy of Greg Ward.)

containing the stroke, and does not appear to make a visual impact on regions previously constrained by other strokes. This is also demonstrated in the video using another image.

Some results of our embedded automatic tone mapper are shown in Figure 6. As explained in Section 4, the resulting tone mapping along with the decomposition of the image into zones provides an excellent starting point for further interactive adjustment. Figure 7 shows two examples produced in this manner. The left column shows the initial automatic tone mapping (top) and the result after interactive adjustment of exposure, contrast, and saturation using a small number of strokes (bottom). Note the enhanced appearance of the sky, mountains, and the green meadow. The right column also shows an initial automatic mapping (top). This time the result (bottom) was produced by adjusting the parameters of several different zones, also resulting in enhanced appearance.

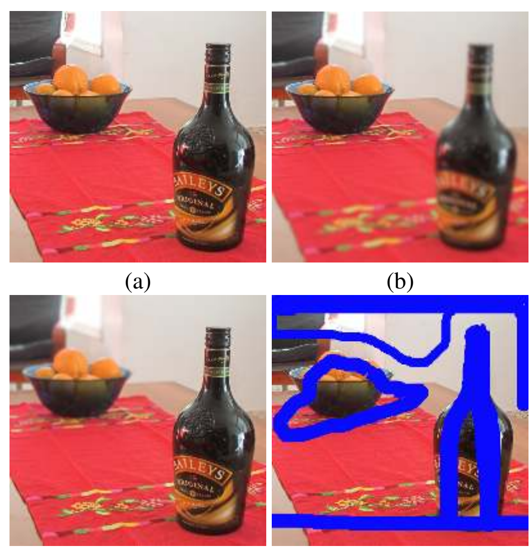
All of the previous examples originated from RAW images or HDR radiance maps. However, our technique can also be an effective tool for quickly touching up ordinary low dynamic range images. Figure 8 shows two examples of such touch ups applied to backlit snapshots. Possibly because of the smaller gradients in these images, a higher value of α (3 instead of 1) was used to produce good results with a small number of brush strokes.

A common case in photography in general, and HDR photography in particular, is that a different white balance adjustment might be required in different image regions. Consider, for example, the scene shown in Figure 9a. Some surfaces in this scene are illuminated by daylight, while others are lit by a much warmer tungsten spotlight. An HDR image of this scene was constructed from several exposures with automatic white balance applied by the camera. In this image, the regions illuminated mostly by the spotlight suffer from a yellowish tint. This problem has been ignored in most previous work on tone mapping (except [Johnson and Fairchild 2003]). With our tool, it is a simple matter to apply a spatially varying white balance correction interactively. The corrected result is shown in Figure 9b. Only two brush strokes were required to perform this local adjustment.

Figure 8: Two examples of touch-ups performed on ordinary low dynamic range images with our tool. The top row shows the original images superimposed with the brush strokes, and the bottom row shows the results after adjustment.

(a) (b)

Figure 9: (a) Image suffering from incorrect while balance (yellow cast in the upper left corner). (b) Applying a spatially varying white balance correction reduces the yellow cast without changing the appearance of the rest of the image.



(a) (b)

(c) (d)

Figure 10: Faking depth of field with spatially varying blur. (a) Original image; (b) the foreground and the far background have been blurred, creating an effect resembling depth-of-field. (c) the foreground is kept sharp, while the background is blurred. (d) The user strokes for these manipulations.

Another interesting application of our tool is to specify a spatially varying degree of blur and/or sharpening across the image. This is demonstrated in Figure 10. In this case the user specifies the σ parameter of the Gaussian blur filter to be applied at each pixel. Three sets of strokes were used to create this result. One set is used to control the blurriness of the bottle and the lower part of the image, another set controls the blurriness of the fruit bowl in the middle, while the third set controls the far background. It should be noted that controlling blur in this manner does not provide a general purpose tool for adding true depth-of-field effects to sharp images. The latter is a challenging problem, which requires the knowledge of accurate per-pixel depth, and as such is outside the scope of our current work.

6 Conclusion

We have described a new interactive tool for locally adjusting tonal values in digital photographs. The tool has been designed with the creative workflow of digital photographers in mind. By combining a simple and intuitive stroke-based user interface with efficient numerical methods we enable local manipulations of images at interactive rates. We also introduced a new automatic tone mapping algorithm, inspired by the Zone System, which compares favorably with existing alternatives.

Time will tell whether the stroke-based interface will become the preferred means for performing local adjustments. In any case, we have demonstrated that the image-guided energy minimization framework provides a useful and general mechanism for constraint propagation, applicable to a variety of adjustments commonly performed in the digital darkroom. In principle, it may be applied to generate spatially variant adjustment maps for any pixelwise operator that may be encoded using a small number of parameters, provided that interpolation of these parameters makes sense.

In future work, we would like to experiment with additional modes of user interaction. In particular, we would like to update the preview *while* the user is drawing the strokes, and to develop edge-sensitive dodge and burn brushes. Another direction for future work is to explore which additional image manipulations might benefit from a tool such as ours. In particular, we would like to develop a more principled approach towards simulating limited depth of field from sharp photographs.

Acknowledgments: Much of this work was carried out while the first author was visiting Microsoft Research. The authors would like to thank Maneesh Agrawala, Michael Cohen, Brian Curless, and David Salesin for their help at the initial stages of this project, and the anonymous reviewers for their thoughtful comments. Thanks also to Norman Koren for letting us use his beautiful images and to Scott Kirkpatrick for helpful discussions and feedback. This work was supported in part by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities and by the Israeli Ministry of Science and Technology.

References

ADAMS, A. 1981. *The Negative*. The Ansel Adams Photography series. Little, Brown and Company.

ADAMS, A. 1983. *The Print*. The Ansel Adams Photography series. Little, Brown and Company.

ADOBE SYSTEMS, INC. 2005. *Adobe Photoshop CS2*. Adobe Systems, Inc., San Jose, CA.

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. *ACM Trans. Graph. 23*, 3, 294–302.

ARTUSI, A., BITTNER, J., WIMMER, M., AND WILKIE, A. 2003. Delivering interactivity to complex tone mapping operators. In *Rendering Techniques 2003*, 38–44.

ASHIKHMIN, M. 2002. A tone mapping algorithm for high contrast images. In *Rendering Techniques 2002*, 145–156.

DRAGO, F., MYSZKOWSKI, K., ANNEN, T., AND CHIBA, N. 2003. Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum 22*, 3 (Sept.), 419–426.

DURAND, F., AND DORSEY, J. 2000. Interactive tone mapping. In *Rendering Techniques 2000*, 219–230.

DURAND, F., AND DORSEY, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph. 21*, 3 (July), 257–266.

FATTAL, R., LISCHINSKI, D., AND WERMAN, M. 2002. Gradient domain high dynamic range compression. *ACM Trans. Graph. 21*, 3 (July), 249–256.

GOODNIGHT, N., WANG, R., WOOLLEY, C., AND HUMPHREYS, G. 2003. Interactive time-dependent tone mapping using programmable graphics hardware. In *Rendering Techniques 2003*, 26–37.

JOHNSON, G. M., AND FAIRCHILD, M. D. 2003. Rendering HDR images. In *IS&T/SID 11th Color Imaging Conference*, 36–41.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. *ACM Trans. Graph. 23*, 3, 689–694.

LI, Y., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2004. Lazy snapping. *ACM Trans. Graph. 23*, 3, 303–308.

LI, Y., SHARAN, L., AND ADELSON, E. H. 2005. Compressing and companding high dynamic range images with subband architectures. *ACM Trans. Graph. 24*, 3, 836–844.

PATTANAIK, S. N., FERWERDA, J. A., FAIRCHILD, M. D., AND GREENBERG, D. P. 1998. A multiscale model of adaptation and spatial vision for realistic image display. In *Proc. ACM SIGGRAPH 98*, M. Cohen, Ed., 287–298.

REICHMANN, M., 2005. An image processing workflow. http://luminous-landscape.com/tutorials/workflow1.shtml.

REINHARD, E., STARK, M., SHIRLEY, P., AND FERWERDA, J. 2002. Photographic tone reproduction for digital images. *ACM Trans. Graph. 21*, 3 (July), 267–276.

REINHARD, E., WARD, G., PATTANAIK, S., AND DEBEVEC, P. 2005. *High Dynamic Range Imaging*. Morgan Kaufmann.

REINHARD, E. 2002. Parameter estimation for photographic tone reproduction. *Journal of Graphics Tools 7*, 1, 45–52.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23*, 3, 309–314.

SAAD, Y. 2003. *Iterative Methods for Sparse Linear Systems*, second ed. SIAM.

SZELISKI, R. 2006. Locally adapted hierarchical basis preconditioning. *ACM Trans. Graph. 25*, 3.

TUMBLIN, J., AND TURK, G. 1999. LCIS: A boundary hierarchy for detail-preserving contrast reduction. In *Proc. ACM SIGGRAPH 99*, A. Rockwood, Ed., 83–90.

WANG, J., AND COHEN, M. 2005. An iterative optimization approach for unified image segmentation and matting. In *Proc. IEEE Intl. Conf. on Computer Vision*, vol. 2, 930–935.

WARD LARSON, G., RUSHMEIER, H., AND PIATKO, C. 1997. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE TVCG 3*, 4, 291–306.