

# Tonal Stabilization of Video

Zeev Farbman  
The Hebrew University

Dani Lischinski  
The Hebrew University



**Figure 1:** Several frames from a video sequence captured by an iPhone. Top row: the in-camera auto white balance causes significant color fluctuations. Bottom row: tonal stabilization eliminates the rapid fluctuations in exposure and color, and the shot may be white-balanced and tonemapped in a consistent manner. **Note:** the video clips for all of the examples in this paper are available on the project web page.

## Abstract

This paper presents a method for reducing undesirable tonal fluctuations in video: minute changes in tonal characteristics, such as exposure, color temperature, brightness and contrast in a sequence of frames, which are easily noticeable when the sequence is viewed. These fluctuations are typically caused by the camera’s automatic adjustment of its tonal settings while shooting.

Our approach operates on a continuous video shot by first designating one or more frames as *anchors*. We then *tonally align* a sequence of frames with each anchor: for each frame, we compute an adjustment map that indicates how each of its pixels should be modified in order to appear as if it was captured with the tonal settings of the anchor. The adjustment map is efficiently updated between successive frames by taking advantage of temporal video coherence and the global nature of the tonal fluctuations. Once a sequence has been aligned, it is possible to generate smooth tonal transitions between anchors, and also further control its tonal characteristics in a consistent and principled manner, which is difficult to do without incurring strong artifacts when operating on unstable sequences. We demonstrate the utility of our method using a number of clips captured with a variety of video cameras, and believe that it is well-suited for integration into today’s non-linear video editing tools.

**Keywords:** tonal alignment, tonal stabilization, color balance, white balance, exposure control, video editing

**Links:**  DL  PDF  WEB

## 1 Introduction

With the proliferation of inexpensive video capturing devices, and the increasing popularity of video sharing websites over the last few years, we have witnessed a dramatic increase in the amount of captured video content. For example, every minute, about 24 hours of video are uploaded to YouTube<sup>1</sup>. Most of this video footage is home-made and captured by amateur videographers using low-end video cameras.

While professional videographers might employ an elaborate setup to control the motion of the camera and the lighting of the scene, home-made video footage often suffers from camera shake and from significant fluctuations in exposure and color balance. These *tonal fluctuations* (seen in the top row of Figure 1) are induced by the camera’s automatic exposure and white balance control: minute adjustments to these tonal settings are continuously made in response to changes in the illumination and the composition of the frame. Turning auto-exposure off is not a practical option, since the dynamic range of the scene is typically much greater than what the camera is able to capture with a fixed exposure setting, making it difficult to avoid over- and under-exposure. Turning off automatic white balance is more feasible, but not all cameras offer this option. In any case, we would like to be able to correct existing videos that were captured with the automatic settings in effect.

While video motion stabilization (elimination of camera shake effects) has been the subject of much research (two recent examples are [Matsushita et al. 2006; Liu et al. 2009]), elimination of tonal fluctuation, or *tonal stabilization*, got surprisingly little attention. In this paper we address this unexplored problem and propose an algorithm for tonal video stabilization.

Different cameras may differ in their response functions, and might employ different auto-exposure and white balance algorithms. Furthermore, a video may have been edited by the user after it has been captured. Therefore, we avoid making strong assumptions regarding the specifics of the camera’s tonal response. Another important feature of our approach is that it does not require computing precise correspondences or accurately tracking features across frames.

<sup>1</sup><http://www.youtube.com/t/press>

Our tonal stabilization method operates on a continuous video shot. One or more frames are designated as *anchors*, typically located in parts of the shot where the tonal settings are stable. Sequences of successive frames are then *tonally aligned* with adjacent anchors: for each frame, we compute an adjustment map that indicates how each of its pixels should be modified in order to appear as if it was captured with the tonal settings of the anchor. This map is efficiently propagated from one frame to the next by taking advantage of temporal coherence. We assume that lighting conditions in the scene are not changing abruptly, and that the tonal fluctuations are of a global nature, rather than spatially varying across the frame.

In order to robustly assess the tonal changes between successive frames we observe that, due to temporal coherence, most of the pixel grid points of any given frame sample the same scene surfaces in the next one. Thus, we have an easily computable set of rough correspondences, making it possible to seed the values of the adjustment map in a large number of locations. Global consistency considerations are then used to propagate these values to the entire frame, obtaining a new complete adjustment map. Thus, the map is propagated between frames, while being gradually updated.

Once a video sequence has been stabilized, it no longer suffers from undesirable fluctuations in exposure and in color. Furthermore, it becomes amenable to a variety of consistent tonal manipulations. For example, the entire sequence can be manually white-balanced by designating a patch in one of the frames as neutral grey. Also, a set of tone curves may be applied to modify the brightness and the contrast of the sequence. The bottom row in Figure 1 shows several frames after applying such corrections on a stabilized sequence. Note that performing such corrections on tonally unstable sequences typically results in artifacts, since the absolute pixel values of the same object in the scene may vary drastically.

## 2 Related Work

Although we are not aware of previous work aimed specifically at tonal video stabilization, a variety of related problems have been well studied in computational photography, image and video processing, and computer graphics.

### Camera response recovery

The digital video capture pipeline may be modeled as follows [Poynton 2003]: the analog linear RGB values arriving at the camera's sensor are converted to digital values, undergo luma/chroma separation, processed to adjust brightness and color, and finally encoded to the target digital video format. Both the analog-to-digital conversion and the subsequent processing may involve non-linear operations. We refer to the combined effect of this pipeline as the camera's response function, which may vary between different cameras operating at different settings, and is typically proprietary.

Had the camera response at each frame been known, it would be possible to stabilize the sequence by inverting the response function. Several methods have been proposed for modeling and recovering the camera response, including parametric [Mann and Picard 1995; Mitsunaga and Nayar 1999; Tsing et al. 2001], semi-parametric [Candocia and Mandarino 2005] and non-parametric [Debevec and Malik 1997] approaches. However, these methods typically operate on still, geometrically registered images, which vary only in their exposure. To apply them to video would require a sufficiently large set of exact correspondences between each pair of frames, which might be difficult to compute. Even if the required correspondences are available, the exposure change between successive frames is typically too small to produce a numerically stable result. Furthermore, it would be necessary to extend these methods to handle more general changes of the camera parameters.

Rather than attempting to recover a full camera response model, our approach uses easily computable rough correspondences between successive frames in order to update a non-parametric model that aligns each frame with the tonal appearance of the anchor.

### Color transfer

At first glance it might seem that tonal alignment may be achieved simply by transferring color from the anchor to the remaining frames. Indeed, a variety of color transfer methods have been proposed over the years [An and Pellacini 2010]. Following Reinhard et al. [2001], several researchers tried to match various global color statistics of two images, such as mean and variance in some color space. Such methods cannot be used for tonal stabilization, since the statistics of a frame tend to vary significantly due to camera and object motion. These changes can occur quite quickly, and therefore any attempts to match the global statistics would result in introducing fluctuations of their own. Local methods, such as [Tai et al. 2005] and [Kagarlitsky et al. 2009] try to find a local match between regions in the image and fit a corresponding offset. While such transfer models are powerful, reliably matching regions in the presence of camera and scene motion remains a challenging task.

Another significant problem in using both global and local methods in the context of frame-to-frame color transfer is that of error accumulation. We further discuss this problem in Section 3.2.

Some recent works [Levin et al. 2004; Li et al. 2008] showed impressive recoloring results using a scribble interface. An and Pellacini [2010] employ this interface in a system specifically designed for user-controlled color transfer between images, using a powerful nonlinear parametric transfer model. We show that it is possible to achieve tonal stabilization using a simpler transfer model, and without requiring user interaction.

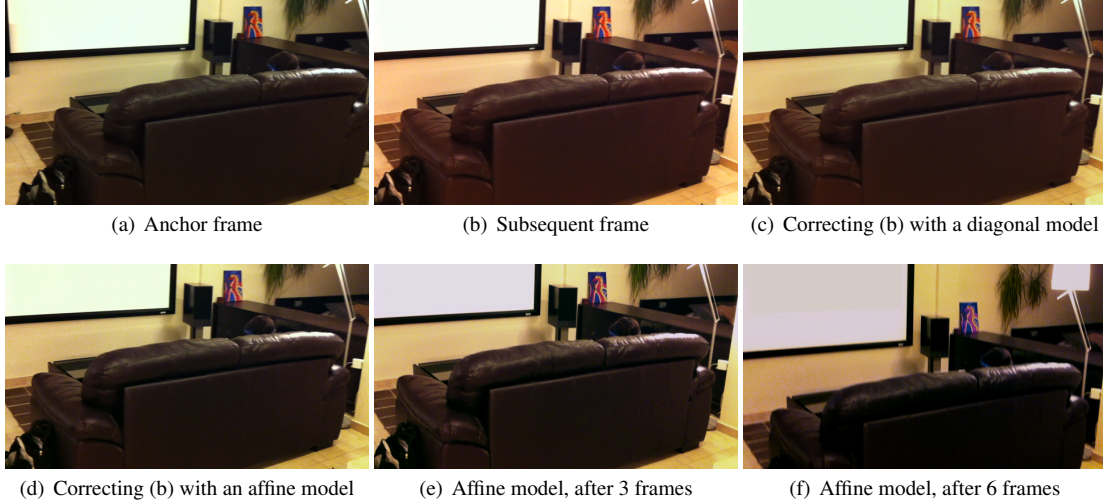
### Color constancy and white balance

Many algorithms for white balance have been proposed over the years; see [Agarwal et al. 2006; Hordley 2006] for a good overview. Most white balance methods relate between pixel values and certain scene attributes, such as the average reflectance, or the illuminant color. Thus, applying such a method on a frame-by-frame basis produces tonal fluctuations of its own, since statistical estimates of the relevant scene attributes typically deviate between frames (causing the tonal fluctuations in the first place). It is also worth mentioning that white-balance corrections are orthogonal to some global color manipulations, such as saturation, and thus cannot undo them. In Section 4 we show that applying a grey-world/grey-edge type white balance algorithms to a tonally-stabilized sequence yields better results than applying them on the original unstable sequence.

### Commercial tools

We are not aware of commercial tools specifically geared at correcting tonal fluctuations in video. Non-linear video editing and post-production tools, such as Adobe Premiere, After Effects and Final Cut Pro, allow to specify tonal corrections at various keyframes and smoothly interpolate them through the entire shot. Thus, in order to stabilize a sequence multiple keyframes must be placed, such that the interpolated corrections would match the underlying fluctuations. Since a video might exhibit many fluctuations over a short period of time, this is potentially a very tedious task.

In the context of time-lapse photography or stop motion animation, there are number of *deflicker* tools, such as GBDeflicker (by Granite Bay Software). While these tools are successful in removing high-frequency fluctuation of exposure, they typically operate on a fixed viewpoint of basically the same scene, unlike the more general sequences we are dealing with in this paper.



**Figure 2:** Parametric estimation of tonal fluctuations. In order to get (e) we composite the affine transformations obtained for 3 successive frame pairs (and 6 transformations for (f)).

### 3 Tonal Alignment

#### 3.1 Overview

The core of our method is the tonal alignment algorithm which, given an anchor frame, adjusts a sequence of adjacent frames to appear as if they had been shot under the same tonal settings. Typically, the frames between a pair of successive anchors are aligned to each one of them, and the final result is obtained by blending the two resulting sequences according to the distance of each frame from each of the anchors. For two anchors that were captured with similar tonal settings, this results in a more error-tolerant alignment of the maps along the frame sequence. More formally, we seek a method for computing  $A_{i+1}$ , given  $f_i$ ,  $A_i$ , and  $f_{i+1}$ .

For each frame  $f_i$ , we compute an adjustment map  $A_i$ . This map specifies for each pixel how its color channels should be adjusted to obtain the desired aligned value. Once we have the adjustment maps, the aligned sequence is obtained simply by applying each map to its frame. Since we know the map at the anchor (the identity mapping), our goal is a method for efficient and robust propagation of the maps along the frame sequence. More formally, we seek a method for computing  $A_{i+1}$ , given  $f_i$ ,  $A_i$ , and  $f_{i+1}$ .

An overview of our method is depicted in the diagram in Figure 3. We make the key observation that due to high temporal coherence of video, most of the pixel grid points of any given frame  $f_i$  sample the same scene surfaces in frame  $f_{i+1}$ . Thus, we compute a set of rough correspondences, which we refer to as the *robust set*  $R_{i/i+1}$ , and use them to seed the adjustment values in a large number of locations.  $A_{i+1}$  is then completed via scattered data interpolation in color space. Before going further into the details of our approach, we would like to discuss some assumptions and considerations that led us to address the problem the way we did.

#### 3.2 Tonal Transformation Model

We assume that the camera’s tonal transformations are global and reversible. Note that not every global transformation can be modeled by independent mappings of the color channels. For example, when the saturation of an image is adjusted, or a general (non-diagonal) white balance correction matrix is applied, the value of each channel in the resulting image depends on all three color channels of the original image, an effect that can only be approximated

with three independent curves. Thus, similarly to previous (still image) color matching methods, our focus at the initial stages of this work was on finding a sufficiently expressive tonal adjustment model, facing a common model selection dilemma: generalization versus overfitting [Bishop 2007].

The inadequacy of a simple, channel independent, model for the task of tonal alignment is demonstrated in Figure 2. A diagonal matrix (three scaling factors, one for each channel) was fit to model the change between images (a) and (b) using a set of user indicated correspondences. This model fails to align image (b) to (a) in a satisfactory manner, as may be seen in image (c), which exhibits some color differences compared to (a) (on the screen and the sofa).

A richer model, which interleaves the channels and has more parameters, such as the one proposed by An and Pellacini [2010], can better account for the variations in the data. This is demonstrated in the second row of Figure 2, where we used an affine transformation with 12 degrees of freedom. Image (d) shows a successful alignment of image (b) to image (a). The difficulty with using high-dimensional models for our purposes, however, is that they tend to overfit the data, and accumulate errors at each stage. Thus, accumulating (compositing) the transformations from one frame to the next results in a rapid degradation in the quality of the result, making it impossible to apply to sequences containing hundreds of frames, as demonstrated in Figure 2(e) and (f).

During the initial stages of this work, we spent considerable time trying to overcome these error accumulation issues. While Kalman filtering and exponential decay weighting [Simon 2006] alleviated error accumulation to some degree, we were not able to produce a stable estimation on a scale of hundreds of frames, needed in the context of tonal alignment.

Eventually, we gravitated towards representing the changes by an adjustment map, defined as a set of three per-pixel additive corrections, or *offsets*. For each frame, we first compute the luminance and then use it to normalize the RGB channels. Thus, we separate between luminance and chroma, and represent the change at each pixel as one offset for the log-luminance channel, and two offsets for the chroma. Note that because we operate on the log-luminance channel, the corresponding offset actually represents an exposure correction. This non-parametric model is expressive by construction, but we shall see in Section 5 that under adverse conditions it

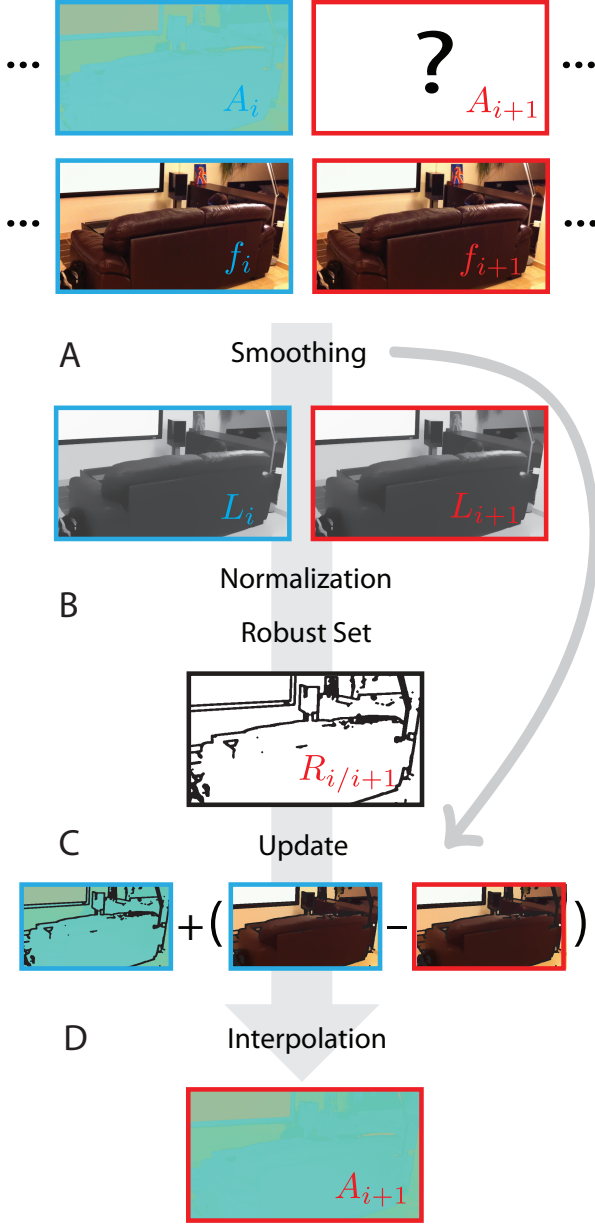


Figure 3: Flow of the adjustment map update algorithm.

gracefully degrades to the equivalent of a much simpler and more limited model, which does not accumulate errors as fast.

### 3.3 Adjustment Map Update

In this section we describe in more detail our algorithm for computing the adjustment map  $A_{i+1}$  given the frames  $f_i$  and  $f_{i+1}$ , and the previous adjustment map  $A_i$ .

Given a pair of corresponding pixels between two frames, any difference between their colors may be attributed to several factors. A change in the tonal parameters of the camera is but one of these factors; the other major factors include changes in the diffuse and specular shading components. Ideally, our goal is to construct adjustment maps that reflect only those color changes that arise from varying the tonal settings of the camera. Previous work has shown that edge preserving smoothing effectively attenuates specularities,

as well as variations in diffuse shading and texture (e.g., [Oh et al. 2001; Khan et al. 2006]). Thus, our first step is to apply a bilateral filter [Tomasi and Manduchi 1998] to each frame (Figure 3A), using the a standard set of parameters: spatial sigma is 10% of the smaller image dimension and range sigma is 10% of the values range.

Next, we efficiently compute a set of correspondences between the successive frames. We rely on the observation that, due to both spatial and temporal coherence, a large set of pixels in two successive frames are likely sample the same surfaces in the scene. We refer to these pixels as the *robust set* (shown in white in Figure 3B).

More precisely, let  $L_i$  and  $L_{i+1}$  denote the luminance channel of the smoothed frames  $f_i$  and  $f_{i+1}$ , with  $\mu(L)$  indicating the mean of the luminance channel. We define the *robust set*  $R_{i/i+1}$  as the set of all pixels whose values in  $L_i$  and  $L_{i+1}$  differ by only a small amount:

$$R_{i/i+1} = \{x \text{ s.t. } |(L_i(x) - \mu(L_i)) - (L_{i+1}(x) - \mu(L_{i+1}))| < 0.05\} \quad (1)$$

The underlying assumption of eq. (1) is that tonal fluctuations in the luminance channel can be approximated by a single shift parameter. All the remaining pixels (whose luminance changed by more than 0.05) are considered likely to have been affected by factors other than a change in the camera’s tonal settings, such as a change in the surface visible through the pixel.

Having computed the robust set  $R_{i/i+1}$ , we use it to initialize the adjustment map at these pixels, while temporarily assigning a value of 0 to the remaining pixels:

$$\hat{A}_{i+1}(x) = \begin{cases} A_i(x) + (f_i(x) - f_{i+1}(x)), & \text{for each } x \in R_{i/i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

In other words, we add the observed color difference at each pixel in the robust set to its previous value in  $A_i$  (Figure 3C).

Next, to obtain  $A_{i+1}$ , we must replace the missing values (zeros) in  $\hat{A}_{i+1}$ . Since we assume that tonal fluctuations are *global* transformations, we want pixels with similar colors in  $f_{i+1}$  to be assigned similar adjustments values in  $A_{i+1}$ , regardless of their spatial location. To achieve this, we employ a fast scattered data interpolation scheme in *color space*. This scheme is derived from Shepard’s method [Shepard 1968] as described below.

The value predicted at pixel  $x$  by Shepard’s interpolation may be expressed as the following weighted sum of values in  $\hat{A}_{i+1}$ :

$$A_{i+1}(x) = \frac{\sum_{r=0}^N w(x, x_r) \hat{A}_{i+1}(x_r)}{\sum_{r=0}^N w(x, x_r) \chi_{\hat{A}_{i+1}}(x_r)} \quad (3)$$

where  $\chi_{\hat{A}}$  is the characteristic function corresponding to  $\hat{A}$  ( $\chi_{\hat{A}}$  is 1 where  $\hat{A}$  is non-zero, 0 otherwise) and  $w$  is a Gaussian distance function (affinity kernel) in color space:

$$w(x, y) = \exp(-\|c(x) - c(y)\|^2 / 2\sigma_c^2). \quad (4)$$

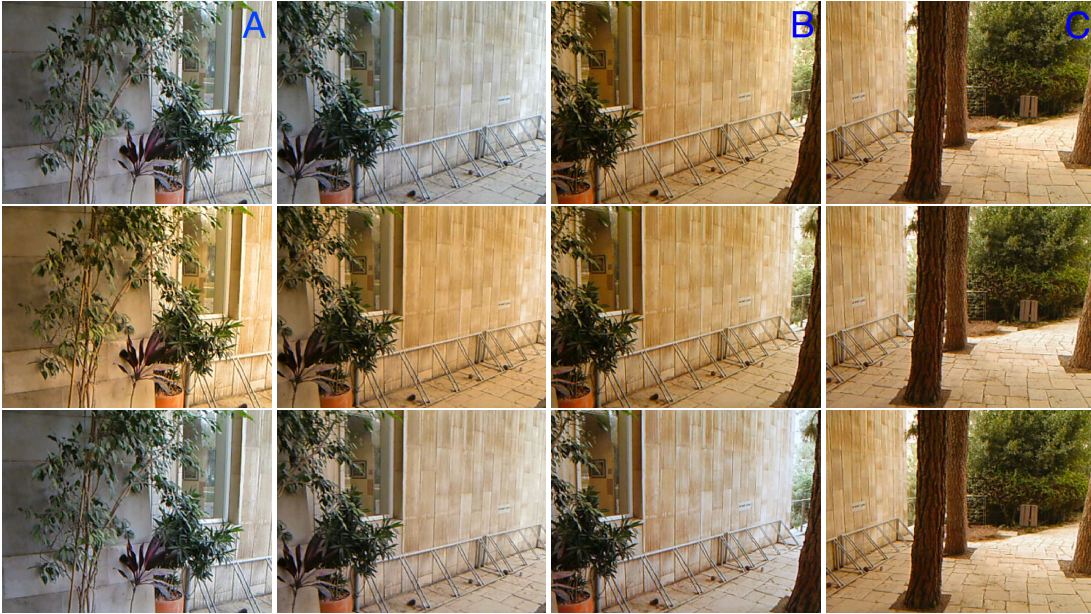
Here,  $c(x)$  and  $c(y)$  are the colors of pixels  $x$  and  $y$  in the CIE Lab color space.

Denoting by  $W$  the all-pairs affinity matrix,  $W_{ij} = w(x_i, x_j)$ , eq. (3) can be rewritten as a ratio of matrix-vector products:

$$\mathbf{A}_{i+1} = \frac{W \hat{\mathbf{A}}_{i+1}}{W \chi_{\hat{\mathbf{A}}}}, \quad (5)$$

where  $A_{i+1}$ ,  $\hat{A}_{i+1}$  and  $\chi_{\hat{A}}$  are represented as column vectors.

Because of the size of  $W$  ( $N \times N$ , where  $N$  is the number of pixels), direct evaluation of eq. (5) is very expensive, but we can use the



**Figure 4:** Effect of anchor placement on the stabilization result. Top row: frames from the original sequence. Middle row: alignment is done using the frame marked B as an anchor. Bottom row: choosing two anchors with a different tonal settings (A and C), produces a smooth transition between the settings.

Nyström method to compute an approximation quickly. We note that the affinity matrix  $W$  is symmetric, and thus diagonalizable by orthogonal matrices:

$$W = UDU^T \quad (6)$$

It has also been shown that all-pairs affinity matrices, such as  $W$ , have low rank [An and Pellacini 2008; Farbman et al. 2010]. In other words,  $W$  can be well approximated using a small number of eigenvectors.

Let  $\tilde{D}$  be a diagonal matrix containing all the eigenvalues of  $D$  above a certain threshold. We can approximate  $W$  by:

$$W \approx U\tilde{D}U^T \quad (7)$$

We use the Nyström method for fast calculation of the eigenvectors of  $W$  [Fowlkes et al. 2004]. Evaluation of eq. (5) now boils down to projecting  $\hat{\mathbf{A}}_{i+1}$  and  $\chi_{\hat{\mathbf{A}}}$  onto a small set of eigenvectors (5–10), which correspond to the eigenvalues in  $\tilde{D}$ . Thus, the resulting cost of the interpolation is linear in the number of pixels in the image.

Our method is close in spirit to the scattered data interpolation scheme of [An and Pellacini 2008], but without the computational overhead of computing the Woodbury formula, which they use in order to invert the associated non-homogenous Laplacian system. It also echoes with [Paris and Durand 2006], where the bilateral filter is approximated using the low frequencies of an operator. Despite these similarities, it is important to remind that our affinity kernel operates on color values only and does not have any component of spatial attenuation.

#### Adjustment map upsampling

In practice, in order to decrease the running time of the algorithm, we work with low resolution adjustment maps. In order to avoid blurring artifacts which arise when applying an upsampled version of this map to the full resolution frame, we use our chroma/luminance separation in the following manner: we apply a low resolution adjustment map to a correspondingly downsampled version of the frame’s luminance channel. Next, we fit a piecewise

linear curve to model the resulting changes in the frame’s luminance. Finally, we use the resulting curve to adjust the luminance of the original resolution frame.

As for the chromatic channels of the frame, they are adjusted using an upsampled version of the adjustment map. This produces acceptable results, since the human visual system is more sensitive to high frequencies in the monochromatic channel.

#### Running times

The main computational bottleneck of our method is the scattered data interpolation scheme based on the Nyström method. The highest adjustment map resolution that we used to produce the results for this paper is 320x180. At this adjustment map resolution, and using 100 samples for the Nyström approximation, the computation time to align each (full-resolution, 1280x720) frame is approximately 1.5 seconds on a laptop with 2.4 GHz (540M) Intel Core i5 processor. The scattered data interpolation takes about 0.3 seconds. While this component is fairly optimized in our Matlab implementation, we believe that significant speed-ups can be gained by a more efficient implementation of the rest of the pipeline.

## 4 Results

We start by demonstrating our stabilization results and the effect that the anchor placement has on them. In Figure 4, the top row shows four sample frames from a sequence with an abrupt color transition<sup>2</sup>. If the frame marked B is selected as an anchor, the entire sequence is aligned to match the tonal characteristics of that frame, as shown in the second row of the figure. Alternatively, we may designate two anchors A and C, one on each side of the color fluctuation. Aligning the sequence to each of the two anchors, followed by blending the results produces a stable sequence that avoids the sudden change in color (Figure 4, bottom row).

<sup>2</sup>All test sequences were captured without fiddling with the camera settings. In fact, some of the videos were shot with a consumer device which does not give the user control over such settings.



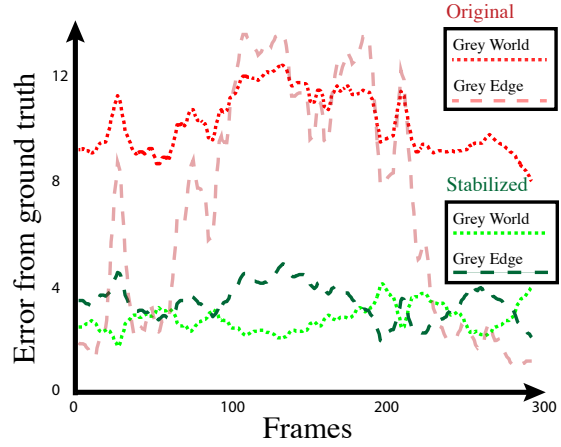
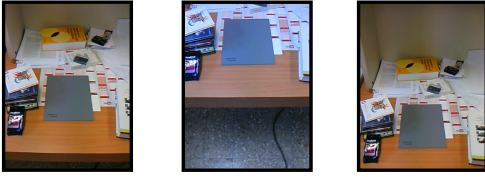
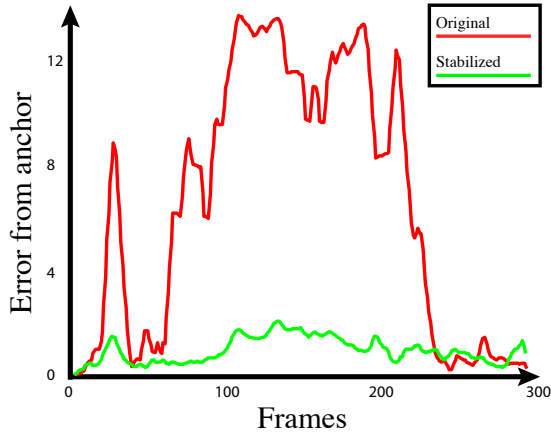
**Figure 5:** Noisy sequence stabilized with an increased affinity sigma  $\sigma_c$ . Top row: original sequence. Bottom row: stabilized sequence



**Figure 6:** Exposure stabilization. Top row: original sequence, showing mainly fluctuations in exposure. Bottom row: exposure was stabilized without changing the chroma. Notice the slight changes in chroma which are still visible after the alignment.



**Figure 7:** Correcting chroma only. In order to avoid excessive clipping, we attenuate only the color shifts of the sequence.



**Figure 8:** Numerical evaluation of tonal stabilization and consequent white balance correction.

The plot on the left shows an angular error (in degrees) between the grey card pixels in each frame and the anchor. The plot on the right shows an angular error (in degrees) from the ground truth.

While exact placement of the anchors plays an important role in the final appearance of the resulting video, generally in order to simply attenuate strong tonal shifts, it is enough to delimit parts of the sequence with strong fluctuations by pairs of anchors. This may be done interactively by the user, or automatically by scanning the sequence for areas where the tonal parameters appear to be stable.

Beyond the attenuation of undesirable tonal fluctuations, our tonal stabilization method offers another significant advantage: once the video is stabilized one can apply a variety of image processing operations in a consistent manner, since color values in each frame now much more closely correspond to values in another frame. In other words, stabilization makes absolute color values to have consistent meaning throughout the sequence. An example of consistently correcting the video using a single tone curve is demonstrated in the second row of Figure 1.

Since our method was particularly motivated by abundance of low-quality video footage from low-end video cameras and mobile devices, it is interesting to see how well our method performs on sequences captured with such devices under adverse conditions, such as scenes with low-level of lighting and high amounts of noise. Figure 5 shows an example of such a noisy sequence. Notice that even in the frames exhibiting very strong amount of noise (which is more evident when viewing the supplementary video), our method succeeds to significantly attenuate the color fluctuations.

In order to achieve satisfactory results on noisy sequences such as the one in Figure 5, or videos of very cluttered scenes, such as the one in Figure 7, we increase the  $\sigma_c$  parameter of eq. (4), because this makes the color affinity function more smooth (and hence more tolerant to noise). Section 5 includes a further discussion of the meaning of this parameter and the trade-offs involved in adjusting it in this manner.

Although so far we presented our algorithm in the context of correcting both chroma and luminance fluctuations, our method allows the user to choose to correct only one of them. For example Figure 6 shows an example where only the exposure was stabilized. Analogously, it is also possible to attenuate only the chromatic fluctuations, while preserving the changes in exposure. This is useful in regions of strong changes in the overall luminance of the scene. Cor-

recting the chroma without attempting to fix the luminance fluctuations allows to avoid the problems caused by the limited dynamic range of the camera. Figure 7 shows an example of chroma only correction, where the magenta color cast in the middle frames of the top row has been removed by the chromatic stabilization.

#### White balance estimation

Once the sequence is tonally stabilized, we can apply a white balance correction in a more principal manner to the entire sequence. We experimented with adapting a Grey-Edge family of algorithms to video. These simple methods, which van de Weijer *et al.* [2007] found to perform well, relative to other alternatives, assume that some property of the scene, such as average reflectance (Grey-World) or average derivative (Grey-Edge) is achromatic.

In order to assess the benefits of tonal stabilization in this context, we have conducted the following experiment. We placed an 18% reflectance grey card in a relatively simple scene and shot the scene with a low-end camera that exhibits strong tonal fluctuations. Next, we aligned the sequence to the first frame by running our stabilization algorithm. In order to quantify the success of the stabilization, for each frame we measured the *angular error* in CIE Lab color space between the mean color of the pixels inside the grey card, to their mean in the first (anchor) frame. The angular error between two colors is defined as the angle between the corresponding color vectors [Hordley 2006].

Figure 8(left) shows that the stabilization process successfully attenuates strong fluctuations in color. The mean angular error before the stabilization is  $6.5^\circ$ , and reduced to  $1.04^\circ$  after the stabilization.

Next, we applied both the Grey-World and Grey-Edge algorithms (with  $\sigma = 1$  and  $p$ -th Minkowsky norm = 2, see van de Weijer *et al.* [2007] for a discussion of these parameters) to both the original and the stabilized sequences. When operating on the stabilized sequence we used the average statistics over all of the frames. When operating on the original sequence, averaging over all frames produced very poor results, and better results were obtained by applying the algorithms to each frame independently, so these are the results we display in Figure 8(right). In this plot we show the angular error with respect to the ground truth color of the grey card.

As can be seen in Figure 8(right), for both Grey-World and Grey-Edge methods, stabilizing the sequence prior to applying the white balancing algorithm greatly improves the results. In the case of the Grey-World algorithm, the mean angular error is reduced from  $10.25^\circ$  to  $2.82^\circ$ . In the case of the Grey-Edge method, the mean error is reduced from  $6.36^\circ$  to  $3.42^\circ$ .

## 5 Discussion

In Section 3.2 we began discussing the trade-offs between different models for fitting the data. While simple models, might not be general enough to capture the space of possible global tonal changes, richer models can easily over-fit the data and accumulate error from frame to frame. The adjustment maps that we chose to use are by definition general enough to represent any tonal change. The question is to what extent they are prone to accumulation of error.

Under adverse conditions, our temporal coherence assumption may be violated. For example, due to strong noise or fast motion, the robust set  $R_{i/i+1}$  might contain only a small fraction of the frame's pixels. Thus, a pixel  $y$  outside the robust set might not be sufficiently similar in color to any of the pixels in the set, and when using scattered data interpolation (eq. (3)) the value of  $A_{i+1}(y)$  will effectively be estimated by averaging rather distant colors.

Intuitively, this means that the adjustment map undergoes diffusion, as it is updated from one frame to the next, rather than being crisply interpolated. Consequently, performing a number of update cycles under adverse conditions will cause the accumulated transformations to lose localization in color space.

Besides the conditions in the video, the exact rate of the diffusion is determined by the affinity Gaussian parameter  $\sigma_c$ , which is the only tunable parameter in our method. For noisy, cluttered sequences, and fast moving sequences, where estimates of the robust set are likely to contain errors, we increase this parameter, creating wider (more diffused) kernels in each row of  $W$ . At the limit, for an extremely high  $\sigma_c$ ,  $W$  degenerates to a constant matrix, and the evaluation of eq. (5) boils down to computing an average per-channel offset. Thus, our color transformation degenerates into a simple 3-parameter model.

In summary, our adjustment scheme is akin to a self-tuning model. It captures the fluctuations of the data at each stage and retains them as far as conditions allow. The diffusion slowly causes accumulated transformations to lose localization in color space and thus become more conservative, avoiding rapid accumulation of error.

While in principle it may be possible to find a concise parametric model to fit the camera tonal variations data, or to devise a rich parametric model that will gracefully degrade into a simpler one once errors begin to accumulate, we leave this avenue of exploration to future work.

### 5.1 Limitations

*Loss of temporal coherence:* As discussed above, when temporal coherency is lost or severely degraded, the algorithm is not expected to behave correctly, since there is not reliable way to update the adjustment for the next frame.

*Changes in lighting vs. exposure settings:* Our framework cannot discriminate between moderate changes in the illumination in the scene, and the gradual changes that occur due to changes in the camera's settings. Thus, our method will attenuate these lighting changes, along with the undesired tonal fluctuation. In practice, this nevertheless tends to make video sequences look better. We leave a more principled identification of the source of changes in brightness to future work.

## 6 Summary

We have presented a method for attenuating undesirable tonal fluctuations in video without resorting to precise tracking or optical flow computations. To our knowledge, this is the first attempt to address this problem, which has important practical applications.

A natural direction for future exploration is to modify our method to handle spatially variant transformations (local edits). Such ability would enable us not only correct local tonal adjustments (which we believe are quite rare in video), but more importantly to propagate local edits in a consistent manner from an anchor to the subsequent frames. While we experimented with propagation of local edits by augmenting the affinity kernel with spatial coordinates, we leave a full exploration of this direction to future work.

**Acknowledgments:** The authors would like to thank Gilad Freedman for many fruitful discussions during the work on this paper. This work was supported in part by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities.

## References

- AGARWAL, V., ABIDI, B., KOSCHAN, A., AND ABIDI, M. 2006. An overview of color constancy algorithms. *JPRR* 1, 1, 42–54.
- AN, X., AND PELLACINI, F. 2008. AppProp: all-pairs appearance-space edit propagation. *ACM Trans. Graph.* 27, 3, Article 40.
- AN, X., AND PELLACINI, F. 2010. User-controllable color transfer. *Comput. Graph. Forum* 29, 2, 263–271.
- BISHOP, C. M. 2007. *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. 2006. corr. 2nd printing ed. Springer, October.
- CANDOCIA, F. M., AND MANDARINO, D. A. 2005. A semi-parametric model for accurate camera response function modeling and exposure estimation from comparametric data. *IEEE Trans. Image Proc.* 14 (Aug), 1138–1150.
- DEBEVEC, P. E., AND MALIK, J. 1997. Recovering high dynamic range radiance maps from photographs. In *Proc. ACM SIGGRAPH 97*, T. Whitted, Ed., 369–378.
- FARBMAN, Z., FATTAL, R., AND LISCHINSKI, D. 2010. Diffusion maps for edge-aware image editing. In *ACM SIGGRAPH Asia 2010 papers*, ACM, New York, NY, USA, 145:1–145:10.
- FOWLKES, C., BELONGIE, S., CHUNG, F., AND MALIK, J. 2004. Spectral grouping using the Nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 2, 214–225.
- HORDLEY, S. 2006. Scene illuminant estimation: Past, present, and future. *Color Res. Appl.* 31, 4, 303–314.
- KAGARLITSKY, S., MOSES, Y., AND HEL-OR, Y. 2009. Piecewise-consistent color mappings of images acquired under various conditions. In *Proc. ICCV*.
- KHAN, E. A., REINHARD, E., FLEMING, R. W., AND BÜLTHOFF, H. H. 2006. Image-based material editing. *ACM Trans. Graph.* 25, 3 (July), 654–663.
- LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. *ACM Trans. Graph.* 23, 3, 689–694.
- LI, Y., ADELSON, E. H., AND AGARWALA, A. 2008. Scribble-Boost: adding classification to edge-aware interpolation of local image and video adjustments. *Computer Graphics Forum* 27, 4 (June), 1255–1264.
- LIU, F., GLEICHER, M., JIN, H., AND AGARWALA, A. 2009. Content-preserving warps for 3D video stabilization. *ACM Trans. Graph.* 28 (July), 44:1–9.

- MANN, S., AND PICARD, R. W. 1995. On being ‘undigital’ with digital cameras: Extending dynamic range by combining differently exposed pictures. In *Proceedings of IS&T*, 442–448.
- MATSUSHITA, Y., OFEK, E., GE, W., TANG, X., AND SHUM, H.-Y. 2006. Full-frame video stabilization with motion inpainting. *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (July), 1150–1163.
- MITSUMAGA, T., AND NAYAR, S. 1999. Radiometric self calibration. In *Proc. IEEE CVPR*, 374–380.
- OH, B. M., CHEN, M., DORSEY, J., AND DURAND, F. 2001. Image-based modeling and photo editing. In *Proc. ACM SIGGRAPH 2001*, ACM, E. Fiume, Ed., 433–442.
- PARIS, S., AND DURAND, F. 2006. A fast approximation of the bilateral filter using a signal processing approach. In *Proc. ECCV*, IEEE, IV: 568–580.
- POYNTON, C. 2003. *Digital Video and HDTV Algorithms and Interfaces*, 1 ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comput. Graph. Appl.* 21, 5, 34–41.
- SHEPARD, D. 1968. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, ACM, New York, NY, USA, ACM ’68, 517–524.
- SIMON, D. 2006. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons Inc.
- TAI, Y., JIA, J., AND KEUNG TANG, C. 2005. Local color transfer via probabilistic segmentation by expectation-maximization. In *Proc. IEEE CVPR*, 747–754.
- TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Proc. ICCV ’98*, IEEE, 839–846.
- TSIN, Y., RAMESH, V., AND KANADE, T. 2001. Statistical calibration of CCD imaging process. In *Proc. ICCV*, vol. I, 480–487.
- VAN DE WEIJER, J., GEVERS, T., AND GIJSENIJ, A. 2007. Edge-based color constancy. *IEEE Trans. Image Proc.* 16, 9 (Sept.), 2207–2214.