MOTION DRIVEN TONAL STABILIZATION

Oriel Frigo[*,†], *Neus Sabater*[*], *Julie Delon*[†] *and Pierre Hellier*[*]

[*] Technicolor, R&I, Rennes, France
[†] Université Paris Descartes, MAP5, Paris, France

ABSTRACT

In this work, we present a fast and parametric method to achieve tonal stabilization in videos containing color fluctuations. Our main contribution is to compensate tonal instabilities with a color transformation guided by dominant motion estimated between temporally distant frames. Furthermore, we propose a temporal weighting scheme, where the intensity of tonal stabilization is directly guided by the motion speed. Experiments show that the proposed method compares favorably with the state-of-the-art in terms of accuracy and computational complexity.

***Index Terms*—** tonal stabilization; white balance; exposure control; video editing;

1. INTRODUCTION

Automatic white balance and automatic exposure are common features of consumer digital cameras. However, these features are not stable in time, and tend to create tonal and brightness instabilities when shooting videos. While these automatic corrections can be turned off in some cases, most cameras offer no control over setup parameters. In this case, the only alternative to avoid unpleasant fluctuations is to further process the video. This preprocessing can also be crucial for computer vision applications relying on brightness or tonal constancy assumptions.

According to [1], the output image u of a camera is related to the irradiance vector $\boldsymbol{e}$ measured by the sensor (RAW[1] intensities) by the relation

$$u = f \circ h(\boldsymbol{T_s} \boldsymbol{T_w} \mathbf{e}) \tag{1}$$

where $\boldsymbol{T_s}$ is a 3×3 matrix transformation that accounts for color space conversion, $\boldsymbol{T_w}$ is a diagonal 3×3 matrix accounting for white balance, $h : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ is a nonlinear gamut mapping operator and $f : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ is the nonlinear tone mapping of the camera. Automatic white balance modifies the matrix T_w, while automatic exposure can be modeled by a multiplicative factor on the vector $\mathbf{e}$, which means that these fluctuations can be considered as **global transformations** on each frame.

[1]Following [2], we consider RAW as the sensor responses after demosaicing step.

If the nonlinear mapping $f \circ h$ was known, it would be possible to invert this function and to correct the new sequence $\{(f \circ h)^{-1}(u_t)\}_t$ with linear transformations. Unfortunately, recovering the camera response function f and the gamut mapping h requires registered images under multiple exposures [3] and a training set of RAW-sRGB pairs [1], which are generally not available for consumer video cameras. Since automatic white balance in cameras is the cause of tonal instabilities, recovering the scene illuminant in each frame of the sequence is also prone to fail in practice. As we will see, this is true even if the available computing power makes it possible to use more involved white balance methods than those used directly in the camera (see [4] for an extensive survey on computational color constancy). The direction we choose in this paper is blind stabilization: the frames of the sequence are corrected together, without any assumption about the scene illuminant and no estimation of the camera response model.

While very few works have attempted to correct color fluctuations in videos, several approaches have been proposed to remove high frequency brightness fluctuations, also known as flicker. These corrections work either globally [5, 6] or locally [7, 8]. Extending flicker correction to color sequences is far from obvious. A possibility is to draw on color transfer, which aims at modifying the colors of an input image according to the colors of an example image. Popularized by [9], global transfer between color distributions has recently evolved in more involved approaches [10, 11]. Local color transfer methods rely on spatial correspondences to derive a color transformation between two images [12]. In the same way, the tonal stabilization presented in this paper draws on a first raw motion estimation between frames to compute a global color correction.

To the best of our knowledge, only two previous works [13, 14] have proposed approaches specifically designed to correct color fluctuations in videos. The first one, due to Farbman *et al* [13], aligns tonally each frame of the video with one or several pre-selected reference frames (for instance the first frame of the movie). Correspondences between successive frames are computed without explicit motion compensation, the authors claiming that many pixels in the same grid position in two successive frames are likely to correspond to the same surface in the scene. If this method provides a reason-

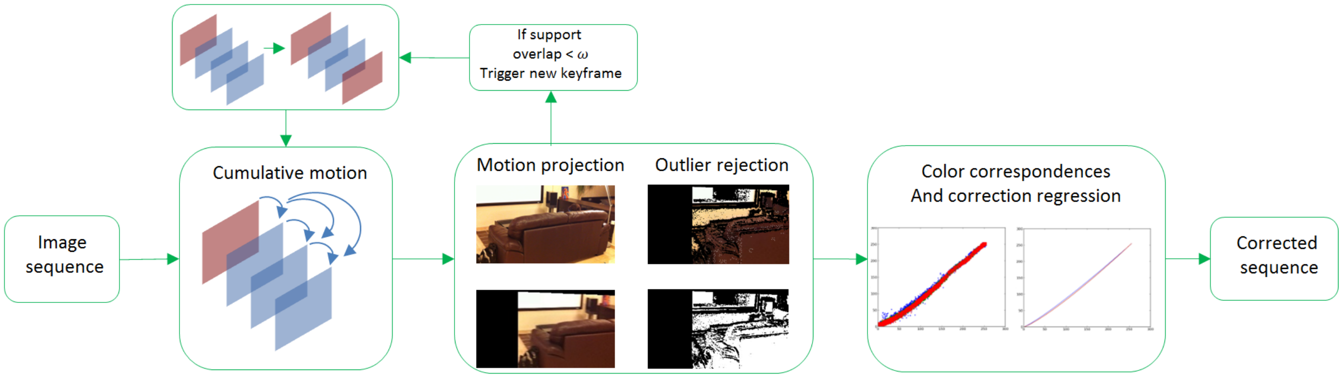


Fig. 1. Overview of proposed method for tonal stabilization of videos.

able solution to stabilize tonal fluctuations in static videos, it is at the cost of high space and time complexity. Besides, the lack of real motion estimation makes the approach unstable in longer, noisy or fast motion sequences.

The second one, recently proposed by Wang *et al.* [14], first estimates global motion between successive frames by relying on local features correspondences. A parametric color transformation is then computed for each pair of neighboring frames. These transformations are accumulated to obtain a color state for each frame of the video, and smoothed along the time direction. While the results provided by the authors are very good, the method remains complex to implement and is based on the choice of several parameters.

Our method, presented in Section 2, relies on a simplified model of color instabilities between the frames of a video, robust to motion and occlusion, and easy to implement with potential to real time processing. In Figure 1, we present a general overview of the proposed method for tonal stabilization. In order to achieve robustness against motion and occlusion, we estimate the dominant motion between a reference keyframe u_k and the frame to be corrected u_t. Then, we register these two frames in order to compute color correspondences. Note that by means of cumulative motion, we are able to register u_t and u_k, even if they differ by several frames in time.

Section 3 illustrates the efficiency of the proposed method on different sequences and shows that it compares favorably to the results of [13] and is qualitatively equivalent to [14] at a much reduced computational cost. We recommend the reader to see our supplementary video results at the project website[2].

2. PROPOSED METHOD

This section presents our method to tonally stabilize a video sequence. First, we present our transformation model for sequences without motion. Then, we generalize our method and employ motion estimation to guarantee tonal stabilization.

2.1. Tonal Transformation Model

Let $\{u_t\}_{t=1,\dots,D}$ be a registered sequence of color frames $u_t : \Omega \to \mathbb{R}^3$, where $\Omega \subseteq \mathbb{R}^2$ denotes the spatial domain. The color channels of u_t are written (u_t^R, u_t^G, u_t^B). Let u_k be a reference keyframe in the sequence. In order to tonally stabilize the sequence, for every following frame $u_t, t > k$ we look for a parametric color transformation $T_t : \mathbb{R}^3 \to \mathbb{R}^3$ such that $T_t(u_t) \simeq u_k$. We use a deliberately simplified model for T_t and assume a separable transfer function on color channels:

$$T_t = (T_t^R, T_t^G, T_t^B), \quad \text{where } \; T_t^c(s) := \alpha_c s^{\gamma_c}, \;\; c \in \{R, G, B\}. \tag{2}$$

In practice, this model is accurate for perceptually correcting tonal instabilities in videos. This model was tested on several image pairs of the same scene, with varying white balance and exposure value. In all cases, this color correction model was accurately capable of correcting tonal variations.

In order to estimate α_c, γ_c, we solve for every color channel of the frame u_t the linear least squares fitting problem

$$\arg\min_{\alpha_c,\gamma_c} \sum_{\boldsymbol{x}\in\Omega} \left(\log u_k^c(\boldsymbol{x}) - \gamma_c \log u_t^c(\boldsymbol{x}) + \log \alpha_c\right)^2, \tag{3}$$

whose solution is given by

$$\gamma_c = \frac{\mathrm{Cov}\left(\log u_t^c, \log u_k^c\right)}{\mathrm{Var}\left(\log u_t^c\right)}, \alpha_c = \exp(\overline{\log u_k^c} - \gamma_c \overline{\log u_t^c}) \,, \tag{4}$$

where $\overline{z} = \frac{1}{|\Omega|}\sum_{\boldsymbol{x}\in\Omega} z(\boldsymbol{x})$ is the average value of $z(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$.

2.2. Image Registration

For all pairs of consecutive frames u_l and u_{l-1} in the sequence, we make use of the robust algorithm of Odobez and

[2] http://oriel.github.io/tonal_stabilization

Bouthemy [15] to estimate the affine motion transformation $A_{l,l-1}$ between the frames. This planar affine transformation, defined by 6 parameters, only accounts for the dominant motion of the camera without considering pixel-wise accuracy. Dominant motion has the advantage of being computationally simple to estimate, and is sufficient for the task of tonal stabilization.

Now, let u_k be a reference frame and u_t ($t > k$) a subsequent frame in the video. Before applying the transformation model in Eq. (7), u_t is warped towards u_k with the accumulated transformation

$$A_{t,k} = A_{t,t-1} \circ A_{t-1,t-2} \circ ... \circ A_{k+1,k}. \tag{5}$$

We define the set of spatial correspondences $\Omega_{t,k}$ between u_k and u_t as

$$\Omega_{t,k} = \Bigg\{ \big(\boldsymbol{x}, A_{t,k}(\boldsymbol{x})\big) \in \Omega \times \Omega \;\Bigg|\; \frac{1}{3}\sum_c \Big((u_k^c(\boldsymbol{x}) \; - \overline{u_k^c}) - (u_t^c\big(A_{t,k}(\boldsymbol{x})\big)) - \overline{u_t^c}) \Big)^2 < \sigma^2 \Bigg\}, \tag{6}$$

where σ^2 is the empirical noise variance (for instance, estimated with the method in [16]). Note that the constraint in Eq. (6) rules out possible motion outliers as well as occluded points (points visible in only one of the frames).

2.3. Algorithm

In practice, the first keyframe of the sequence is the first frame u_1. Then, the next sequence of frames $u_t, t > 1$ are tonally stabilized w.r.t. to u_1 provided the number of spatial correspondences $\#\Omega_{t,1}$ is larger than $\omega \times \#\Omega$ where $\#\Omega$ is the number of pixels per frame, and ω is a parameter to be tuned. Otherwise, since there is not enough overlap between Ω_1 and Ω_t to robustly estimate the tonal transformation, a new keyframe is defined. This process is repeated till the end of the sequence.

In order to ensure that the sequence $\{T_t(u_t)\}_{t=1,...,D}$ does not deviate largely from the original sequence $\{u_t\}_{t=1,...,D}$ we consider the temporally weighted transformation

$$T'_t = \lambda T_t + (1-\lambda) Id, \tag{7}$$

where $\lambda = \lambda_0 \exp(-\frac{||V_{t,k}||}{p})$ regulates the amount of transformation of frame u_t depending on the motion speed in reference to the keyframe, $||V_{t,k}||$ denotes the norm of the dominant motion vector $V_{t,k}$ and p is the maximum spatial displacement (number of rows + number of columns of the image) and λ_0 is the initial weight (in practice we set $\lambda_0 := 0.9$). In the sequel we consider this transformation model since it allows to stabilize the sequence, but avoiding overexposure when huge changes in camera exposure occur in the original sequence.

Algorithm 1 sketches the proposed method. Note that the computation of $\Omega_{t,k}$ involves the computation of $A_{t,k}$ and the warping of u_t towards u_k based on $A_{t,k}$.

Algorithm 1 Motion driven tonal stabilization

```
Input: Sequence of frames {u_t}_{t=1,...,D}
Output: Tonal stabilized sequence {T'_t(u_t)}_{t=1,...,D}
k ⇐ 1, t ⇐ k + 1
T'_1(u_1) = u_1
while t ≤ D do
    Compute Ω_{t,k}
    if #Ω_{t,k} ≥ ω × #Ω then
        for c ⇐ {R, G, B} do
            α_c, γ_c ⇐ argmin_{α,γ} Σ_{(x,y)∈Ω_{t,k}} (u_k^c(x) − α u_t^c(y)^γ)^2
            T'_t(u_t^c) ⇐ λ α_c (u_t^c)^{γ_c} + (1 − λ) u_t^c
        end for
        t ⇐ t + 1
    else    # If there are not enough correspondences
        k ⇐ t − 1
        u_k ⇐ T'_{t−1}(u_{t−1})
    end if
end while
```

For the sake of complexity, the original frames are rescaled (120 pixels wide) before estimating T'. Then, the estimated T' is coded into a Look-Up-Table (LUT) that is applied to the high resolved original frames. This implementation changes guarantee an algorithm with low complexity but it does not produce noticeable loss in tonal stabilization accuracy.

3. EXPERIMENTAL RESULTS

3.1. Qualitative results

Results on a variety of sequences are accessible at the project website[3]. The tonal stabilization method was tested on 18 sequences, originating either from related work [13, 14], or acquired with hand-held smart-phones from different manufacturers. All sequences were processed with the same set of parameters, i.e., $\omega = 0.25$ and $\lambda_0 = 0.9$.

Visual inspection was performed for all sequences and the method has proven to be accurate and robust. Fig. 2 and Fig. 3 show examples of obtained results.

3.2. Quantitative results

Here, the performance of the stabilization is assessed and compared objectively to two existing methods [13, 14]. We implemented to the best of our ability the method of [13], while the authors of [14] provided us with the results. For

[3] http://oriel.github.io/tonal_stabilization

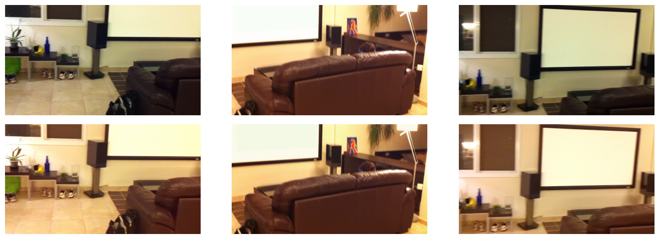
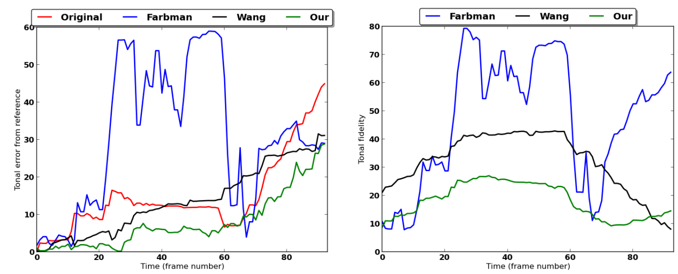


Fig. 2. Illustration of tonal stabilization of sequence "sofa". Top row, frames extracted from original sequence. Second row, same frames, after tonal stabilization with our method. It is visible that objects have the same color appearance through the sequence.

objective assessment, we compute the tonal variation (euclidian distance in CIELAB color space) of a homogeneous patch in comparison to the reference (first frame). Results obtained on the sequence *building* are shown in Fig. 4. We observe that the tonal variation is reduced after correction with our method and with the method of [14] when compared to the original variation. However, the result of [13] presents some flickering. The fidelity to original colors, measured by the tonal variation between the corrected frame and the original frame, can indicate if a method produces noticeable artifacts (large deviation from original).

Fig. 3. Visual comparison of tonal stabilization for sequence "building". It can be observed that the proposed method is able to correctly stabilize tonal variations, while minimizing undesired artifact generation, while the results of [13] and [14] contains some visible clipping in the building.

We note that the proposed tonal stabilization method is much faster in comparison to the state-of-the-art. Our prototype implementation in Python processes a $1920 \cdot 1080$ resolution video in a rate of 11 frames per second[4], while a C++ implementation of [14] processes approximately 1 frame per second and a Python implementation of [13] processes 0.6 frames per second. We believe that an optimized implementation of our method could approach real time processing.



Tonal Variation   Fidelity to Original

Fig. 4. Objective assessment on the sequence *building*, relating to Fig. 3. On the left, the tonal variation over time, computed as the color distance of a tracked patch to the reference (first frame) is shown. On the right, the color distance is computed at each instant, between the corrected frame and the original frame, which indicates the degree of fidelity between the original and the corrected sequence. In particular, for method [14], the whitish appearance of the corrected sequence is assessed by a large color distance. Overall, our method compares favorably with the methods of [13] and [14], both in terms of *reduction of tonal variation* as in terms of *fidelity to original colors*.

4. CONCLUSION

In this work, we have proposed an efficient tonal stabilization method, aided by global motion estimation and a parametric tonal transformation. We have shown that a simple six-parameters color transformation model is enough to provide tonal stabilization caused by automatic camera parameters, without the need to rely on any *a priori* knowledge about the camera model.

The proposed algorithm is robust for sequences containing motion, it reduces tonal error accumulation by means of long-term tonal propagation, and it does not require high space and time computational complexity to be executed.

In addition, one of the main advantages of the proposed method is that it could be applied in practice as an online algorithm, that has potential for real time video processing.

Acknowledgements: We thank the authors of [14] for providing their results in our sequences and [13] for helping with implementation details of their method.

[4]Processing time given by Intel(R) Core(TM) i5-3340M CPU @ 2.70GHz, 8GB RAM

5. REFERENCES

[1] Haiting Lin, Seon Joo Kim, Sabine Susstrunk, and Michael S. Brown, "Revisiting radiometric calibration for color computer vision," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 129–136.

[2] Seon Joo Kim, Hai Ting Lin, Zheng Lu, Sabine Süsstrunk, Stephen Lin, and Michael S. Brown, "A new in-camera imaging model for color computer vision and its application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 2289–2302, 2012.

[3] M Grossberg and S Nayar, "What Can Be Known about the Radiometric Response from Images?," in *Computer Vision - ECCV 2002*, pp. 189–205. 2002.

[4] A. Gijsenij, T. Gevers, and J. Van De Weijer, "Computational color constancy: Survey and experiments," *Image Processing, IEEE Transactions on*, vol. 20, no. 9, pp. 2475 – 2489, 2011.

[5] Étienne Decencière, *Restauration automatique de films anciens*, Theses, École Nationale Supérieure des Mines de Paris, Dec. 1997.

[6] J. Delon, "Movie and video scale-time equalization application to flicker reduction," *Image Processing, IEEE Transactions on*, vol. 15, no. 1, pp. 241–248, Jan 2006.

[7] F. Pitié, R. Dahyot, F. Kelly, and A. Kokaram, "A new robust technique for stabilizing brightness fluctuations in image sequences," in *Statistical Methods in Video Processing*, pp. 153–164. Springer, 2004.

[8] J. Delon and A. Desolneux, "Stabilization of flicker-like effects in image sequences through local contrast correction," *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 703–734, 2010.

[9] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34 –41, sep/oct 2001.

[10] Sira Ferradans, Nicolas Papadakis, Julien Rabin, Gabriel Peyr, and Jean-Franois Aujol, "Regularized discrete optimal transport," in *Scale Space and Variational Methods in Computer Vision*, Arjan Kuijper, Kristian Bredies, Thomas Pock, and Horst Bischof, Eds., vol. 7893 of *Lecture Notes in Computer Science*, pp. 428–439. Springer Berlin Heidelberg, 2013.

[11] O. Frigo, N. Sabater, V. Demoulin, and P. Hellier, "Optimal transportation for example-guided color transfer," in *Proceedings of Asian Conference on Computer Vision*, 2014.

[12] Yoav HaCohen, Eli Shechtman, Dan B. Goldman, and Dani Lischinski, "Non-rigid dense correspondence with applications for image enhancement," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 70:1–70:10, July 2011.

[13] Zeev Farbman and Dani Lischinski, "Tonal stabilization of video," *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2011)*, vol. 30, no. 4, pp. 89:1 – 89:9, 2011.

[14] Y. Wang, D. Tao, X. Li, M. Song, J. Bu, and P. Tan, "Video tonal stabilization via color states smoothing.," *IEEE transactions on image processing*, vol. 23, no. 11, pp. 4838–4849, 2014.

[15] JM. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of visual communication and image representation*, vol. 6, no. 4, pp. 348–365, 1995.

[16] Miguel Colom and Antoni Buades, "Analysis and Extension of the Percentile Method, Estimating a Noise Curve from a Single Image," *Image Processing On Line*, vol. 3, pp. 332–359, 2013.