Pacific Graphics 2017
J. Barbic, W.-C. Lin, and O. Sorkine-Hornung
(Guest Editors)

Volume 36 (2017), Number 7

Photometric Stabilization for Fast-forward Videos

Xuaner Zhang[1,2], Joon-Young Lee[2], Kalyan Sunkavalli[2], and Zhaowen Wang[2]

[1]University of California, Berkeley, USA, [2]Adobe Research, USA

Abstract

Videos captured by consumer cameras often exhibit temporal variations in color and tone that are caused by camera auto-adjustments like white-balance and exposure. When such videos are sub-sampled to play fast-forward, as in the increasingly popular forms of timelapse and hyperlapse videos, these temporal variations are exacerbated and appear as visually disturbing high frequency flickering. Previous techniques to photometrically stabilize videos typically rely on computing dense correspondences between video frames, and use these correspondences to remove all color changes in the video sequences. However, this approach is limited in fast-forward videos that often have large content changes and also might exhibit changes in scene illumination that should be preserved. In this work, we propose a novel photometric stabilization algorithm for fast-forward videos that is robust to large content-variation across frames. We compute pairwise color and tone transformations between neighboring frames and smooth these pair-wise transformations while taking in account the possibility of scene/content variations. This allows us to eliminate high-frequency fluctuations, while still adapting to real variations in scene characteristics. We evaluate our technique on a new dataset consisting of controlled synthetic and real videos, and demonstrate that our techniques outperforms the state-of-the-art.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms I.4.3 [Image Processing and Computer Vision]: Enhancement —Smoothing

1. Introduction

The ubiquity of mobile cameras and video sharing platforms such as Youtube, Instagram, and Snapchat has made video capture and processing extremely popular.

However, while it is easy to capture videos, viewing and sharing long, unprocessed videos is still tedious. A popular way to compress videos into shorter clips is to fast-forward them, and timelapse and hyperlapse are two appealing techniques to accomplish this; the former handles videos captured using static (or slow-moving) cameras over a long period of time (*e.g.*a day-to-night landscape shown in one minute), while the latter is applied to videos captured by moving (often hand-held) cameras that covers large distances (*e.g.*a hike across the Great Wall summarized in one minute). These videos are created by sampling only a subset of the frames (either uniformly or taking video features into account [JKT*15, PHAP15]).

Most videos captured by consumer devices exhibit temporal variations in color and tone that can be caused by either scene changes (*e.g.*variations in scene illumination) and imperfect compensation by in-camera processing such as auto-exposure and white-balance. These photometric fluctuations are particularly troubling when they are high-frequency in nature. This problem is exacerbated in the case of fast-forward videos because frame-sampling changes even low-frequency color and tone variations into bother-

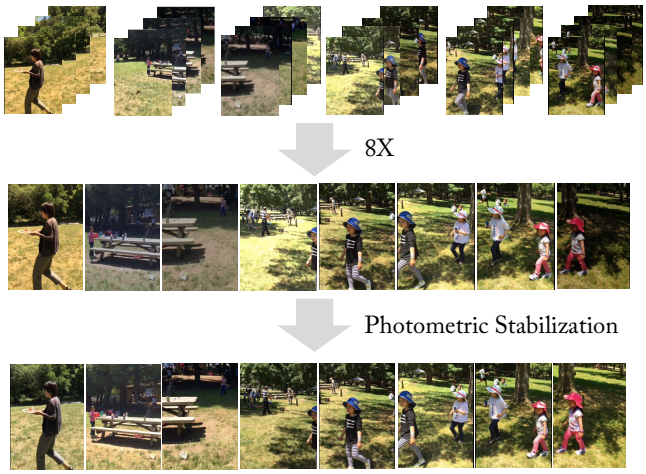

Figure 1: *Photometric instability in fast-forward videos. (TOP) original image sequence with photometric jitter, e.g.brightness and tone fluctuations. (MIDDLE) sampled frames for fast-forward video with a speedup of 8, which exacerbates the photometric variations (BOTTOM) our result video after photometric stabilization.*

some high-frequency flickering (see Fig. 1). Our proposed stabilization framework applies content-aware filtering to fast-forward videos that exhibit undesired photometric jitter and large content variation. We are able to automatically detect and remove high-frequency fluctuations while preserving a smooth scene change.

2. Related Work

Color and exposure fluctuations are common problems in videos and professional video editing software such as Adobe Premiere and Adobe After Effects have tools to rectify them. However, these tools require significant user effort to manually adjust colors frame by frame. Farbman and Lischinski [FL11] proposed a technique to automatically stabilize tonal variations in videos by applying a pixel adjustment map to align all frames to a set of user-selected anchor frames. Their technique computes dense correspondence by assuming small inter-frame motion, and fails on fast-forward videos which can have large motion and content changes. It also requires the selected frame to have good photometric properties to be a reference. Similar reference-based techniques are also presented in [BSPP13, BTS*15], in which they use a video as a filter to transfer the target videos to the same tone and style; in [VCB14], an image is selected as a reference in order to form a consistent set of images taken by various camera sources and settings. In contrast, our method doesn't require a reference video or frame, and is able to handle arbitrary input videos.

Frigo *et al*.. [FSDH15] extended this work by computing global motion, automatically inserting anchor frames in case of large motion, and weighting the correction by the magnitude of motion. While this improves on the previous technique, it has no notion of content similarity and will fail on videos with large content variation. In contrast, our technique computes pair-wise color transformations without requiring dense correspondence, and automatically filters these transformations taking potential content/illumination changes into account.

Wang et al. [WTL*14] recently propose a stabilization technique that computes pair-wise affine color transformations, which they refer to as color states. They compute "absolute" color transformations between the first frame and subsequent frames; this requires long-range feature tracking that can fail on fast-forward videos. They compute PCA to smooth out the entire sequence states in a synchronized manner. However, their use of PCA over all the color states restricts them to work with only short video clips. In addition, they rely on a frame registration step that only works in the presence of small motion. In contrast, our technique depends on purely local (temporal) processing – making it computationally more efficient – and generalizes better to large motion and content change.

3. Photometric Stabilization

Given a sequence of frames that contain large content variation, our goal is to apply photometric (both luminance and chrominance) stabilization that preserves the original scene illumination changes but removes high frequency color fluctuations.

When the frame sequence has fairly little content change, feature tracking based approaches work quite well [FSDH15, GKE11]; however, when there are large content variations even within neighboring frames, as often seen in fast-forward videos, feature tracking is not applicable. Thus we rely on only pairwise transformations between successive frames. We accumulate these transformations using regularization to compute longer range transformations. We then smooth these transformations using a temporally-weighted filter that accounts for photometric and content change as well as outlier frames. To avoid artifacts caused by smoothing correlated transformation parameters, we smooth at pixel (correspondence) level, with which to re-compute the desired transformation. Finally, we apply the difference between the original color transformations and their smoothed counterparts to create the final stabilized video.

In the following section, we first describe how we perform photometric alignment between frames then explain how we achieve photometric stabilization over an entire video.

3.1. Photometric alignment between frames

Given fast-forward videos often have large content variation across neighboring frames, we only calculate the photometric transform between two successive frame pairs. For each pair of frames, we first extract local image features and compute a homography transformation to align the frame pair. We used ORB [RRKB11] feature in all our motion models.

We randomly sample a subset (5% in our implementation) of corresponding pixel values from the aligned frame pair. We denote the set of sampled correspondences between adjacent frames i and $i+1$ as (p_i, q_i). We estimate the pairwise photometric transformation, $T_{i,i+1}$, by minimizing the energy function defined as:

$$\sum_{(p_i,q_i)\in P_i} \|T_{i,i+1}(\theta)p_i - q_i\| + \lambda\|T_{i,i+1}(\theta) - \mathbb{I}\|, \qquad (1)$$

where (p_i, q_i) represents a pair of corresponding pixel values, λ is the weight for regularization, and $\mathbb{I}$ denotes an identity transformation.

Our framework does not place constraints on the choice of the transformation model, T, to use or color space to work with. In our implementation, we consider photometric smoothing in luminance and chrominance channels separately in the decorrelated *YCbCr* color space. For both luminance and chrominance, we model the color transfer as a global transfer, which is able to account for camera auto-adjustment and global scene illumination change.

When source and target image pair has correspondences, it is more precise to calculate color transfer directly using correspondences, instead of indirectly matching color statistics [RAGS01, PKD05, PKD07] or brightness transfer function [KLL*12]. Although we deal with videos with large content variations, we only compute the transfer between a pair of successive frames, thus the amount of correspondences is sufficient to optimize an accurate pairwise transfer model. Specifically, we use the color transfer model as below.

Luminance We found it sufficient to use a simple weighted gamma curve mapping $T(\{\alpha,\gamma\}, Y)$ to model pairwise luminance transfer, which is denoted as:

$$Y^q = \alpha(Y^p)^\gamma \qquad (2)$$

where (Y_q, Y_p) are the luminance values at the corresponding pixels, and $\{\alpha, \gamma\}$ are model parameters. The effectiveness of gamma curve estimation is described in detail in [VCB15].

Chrominance We use a standard 2×3 affine transformation to model the 2 channel chrominance transfer:

$$\begin{bmatrix} C_b^q \\ C_r^q \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} C_b^p \\ C_r^p \\ 1 \end{bmatrix}. \quad (3)$$

We solve for the luminance and chrominance transforms by solving a linear least squares problem (the luminance problem can be made linear by taking log). Since not all correspondences from the alignment are accurate, we add robustness to the step by estimating the parameters via RANSAC. Specifically, for both luminance and chrominance transform, we set maximum iteration of RANSAC to be 1000, and marked a frame as degenerated when the color transform produces inlier fewer than 1% of the shorter side of the image size.

Given all the pairwise transformations, $T_{1,2}, T_{2,3}, ..., T_{N-1,N}$, we can compute the transformation between an arbitrary frame pair i and j, by accumulating transformations between them as $T_{i,j} = T_{j-1,j} \cdots T_{i+1,i+2} T_{i,i+1}$.

However, accumulated transformations can introduce color artifacts (see bottom right of Fig. 2 for an example).

To alleviate such model bias, we accumulate correspondences from neighboring frame pairs, (a proportion of β% where $\beta = \frac{100}{2^{|i-k|}}$ such that $k \in [-5, 5]$). Thus the pixel samples (p_i, q_i) used in Eq. (1) are accumulated correspondences from a window of neighboring frame pairs. Note that computing the transformations from features tracked across frames could have been more accurate, but for fast-forward videos, neighboring frames do not have sufficient correspondences.

Figure 2: *Color transfer by accumulating pairwise transformations. The reference frame is frame i and the target frame is frame $i+10$. The target is matched to the reference using accumulated transformation (in this case, the accumulation of 9 transformation matrices). (TOP) the reference and target frame pair (BOTTOM LEFT) the transformed frame with accumulation (BOTTOM RIGHT) the transformed frame with no accumulation.*

3.2. Photometric stabilization by weighted filtering

After getting pairwise transformation between arbitrary frame pairs, we filter the transformations to create a set of desired smoothly-varying transformations.

While doing this, it is important to account for content of the video. For example, a large variation in the pixel colors might correspond to a high-frequency jitter in the camera white-balance. On the other hand, it might also be a result of real changes in scene content. Our goal is to remove the first, while smoothly retaining the second.

To this end, we need a metric that allows us to distinguish between the two. In order to do this, we propose a photometric similarity measure between two frames, that compares their color distributions using the Earth Mover's Distance (EMD) [RTG00]. Note that earlier when computing pairwise color transfer, we used correspondences-based alignment, which is more robust to outliers (*e.g.*a new foreground object) than histogram transfer. This gives us accurate color transfer but no content information. Here we chose to use histogram-based color comparison between image pairs for similarity measure due to the following two reasons. First, when comparing image pairs with a larger temporal span, correspondence-based approach fails due to the lack of correspondences. Second, histogram-based comparison allows us to infer content variation by comparing color aligned image pairs after correspondences-based alignment.

Using the EMD measure, we define the photometric distance as:

$$\mathbb{D}_{i,j} = EMD(\text{pdf}(p_i), \text{pdf}(q_j)), \quad (4)$$

where $\text{pdf}(p_i)$ and $\text{pdf}(q_j)$ represent the histogram of corresponding pixel values in the frame i and j respectively. Given two frames, we can compute a photometric transformation that aligns the two (as per Sec. 3.1) and then compute the similarity measure. This allows us to eliminate differences that might have been caused by camera adjustments, and then measure content differences.

Given this similarity measure, we now define our weighted smoothing filter. We apply a weighted filter W_i of size M to each frame i, where M is the number of neighboring frames used to correct the target frame. Denote neighboring frames of frame i as $i - M/2, ..., i + M/2$. The overall weight W is composed of four terms: identity weight W_I, temporal weight W_T, content weight

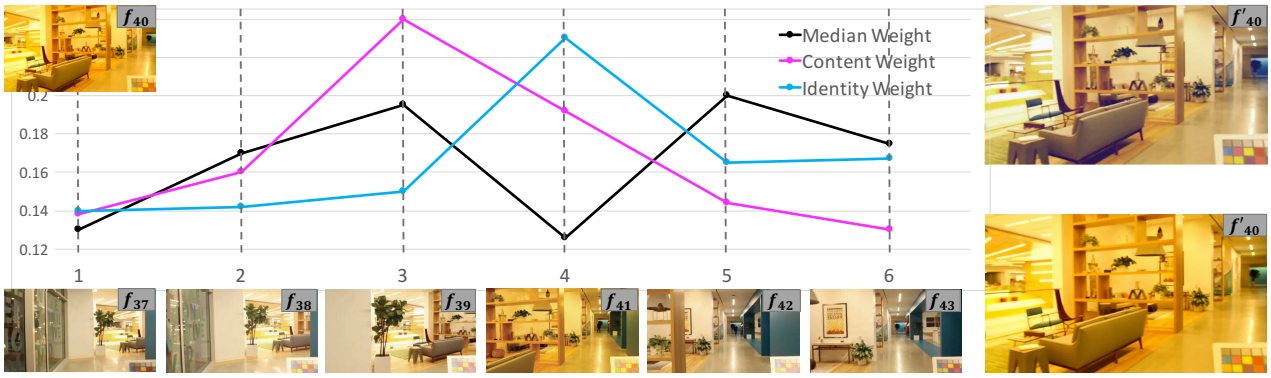

Figure 3: *Illustration of the effect of different weight filters. The target frame is plotted on the upper left corner of the figure with the weight plots for all the 6 neighboring frames of the target frame (LEFT). Notice that both the target and one neighboring frame f_{41} are outliers with sharp tone jitter, and f_{41} also contains similar content with the target frame. This leads to high identity and content-aware weights but low outlier weights for f_{41} (because its colors are dissimilar to the remaining frames in this window). Without the outlier weight, f_{41} would be weighted highly among the 6 neighbors leading to poorer filtering (LEFT, BOTTOM), but accounting for it leads to a temporally smoother result (RIGHT, TOP).*

W_C and outlier weight W_M. W is computed as a normalized sum of all 4 terms: $W(i,j) = \frac{1}{N_I}W_I(i,j) + \frac{1}{N_T}W_T(i,j) + \frac{1}{N_C}W_C(i,j) + \frac{1}{N_M}W_M(i,j)$, where N_I, N_T, N_C and N_M are normalization factors computed such that all the weights are at a similar numerical scale. We chose to scale each weight vector by its median. In the following, we will elaborate on the four weights described above.

(1) Identity weight: The first term is an identity term that penalizes neighboring frames that have different photometric values from the target frame i

$$W_I(i,j) = \exp(-\mathbb{D}_{i,i'}) \tag{5}$$

where i' is the simplified notation of transformed pixel samples of the frame i by applying $T_{i,j}$, i.e, $T_{i,j}(p_i)$. If the color of frame j is similar to that of frame i, the transform between frame i and j should approach identity. Applying this transformation to p_i should produce values that are very similar to p_i. Thus, this metric is a way of evaluating if the neighboring frames are very similar to the current frame (leading to close to identity transformations).

(2) Temporal weight: The second temporal term simply penalizes frames that are temporally far from the target frame i

$$W_T(i,j) = \exp(-((i-j)^2/(2\sigma^2))) \tag{6}$$

(3) Content-aware weight: In order to smooth out high-frequency variations, we would like to average out transformations over frames that are similar in content. To do this, we compute the distance between color aligned images, specifically the transformed sample distribution $\text{pdf}(p'_i) = \text{pdf}(T_{ij}p_i)$ of frame i and the sample distribution $\text{pdf}(j)$ of frame j:

$$W_C(i,j) = \exp(-\mathbb{D}_{i',j}) \tag{7}$$

As noted above, differences between the transformed target distribution and the source distribution would indicate content change between the frame pair like dynamic objects in the scene. On the other hand, a simple camera adjustment would get equalized by the transformation T_{ij} leading to larger similarity and a large filter weight. Note that content-aware weight would be equivalent to identity weight when the scene is static. However, when motion presents, content weight penalizes large content change, even if the color transform is close to identity.

(4) Outlier weight: We define outliers to be frames that contain sharp change in either brightness or color, and should be weighted much less during the computation of reference distribution. However, when computing the filter weights according to measures $(1--3)$, such frames will give their neighboring frames low weights (because of the strong changes in color). Instead, we need to eliminate them from the weighting scheme. We assume the outliers are sparse in the selected frame sequence, and thus a majority vote approach such as median filtering would be effective.

$$W_M(i,j) = \exp(-\|\mathbb{D}_{i,med} - \mathbb{D}_{i,j}\|) \tag{8}$$

where $D_{i,med}$ denotes the EMD distance between the target distribution and the median distribution within its neighboring range. The outlier weight is especially crucial when the target frame is an outlier and there exists other outliers in its neighboring frames.

Fig. 3 demonstrates how all these weights combine to give us smooth, robust, filtering on a video sequence with jitter. We also compared between using the proposed weighted filtering and using naïve uniform filtering; the comparison result is shown in Fig. 4.

3.2.1. Rendering photometrically stabilized frames

Given the transformation weights, we would like to use them to smooth out the photometric variations in the video sequence. One option to do this could be to compute smoothly varying transformations $\hat{T}_i$ by directly applying the weighted filter to the original transforms:

$$\hat{T}_i = \sum_{j=i-M/2}^{j=i+M/2} W(i,j)T_{i,j}. \tag{9}$$

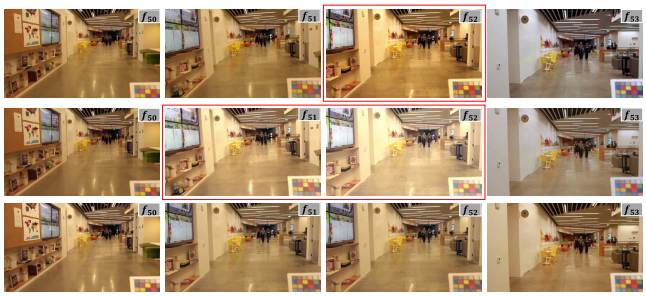

Figure 4: *Illustration on the effectiveness of the proposed weighted filtering. (TOP) input sequence with high frequency photometric jitters (e.g.the $f_{52}(3^{rd})$ contain brightness jitter, and $f_{53}(4^{th})$ frame contains color fluctuation); (MIDDLE) result of uniform filtering, note that frames with jitter cannot be corrected, and can even affect neighboring frames that are originally correct (e.g.the $f_{51}(2^{nd})$ frame is affected by $f_{52}(3^{rd})$ frame to appear brighter) (BOTTOM) result using the proposed content and outlier-aware weighted filtering; the frames can be corrected properly.*

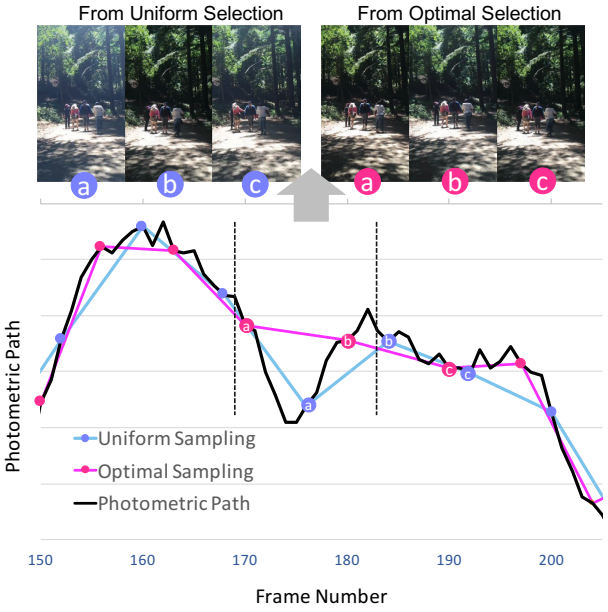

Figure 5: *Uniform selection VS. optimal frame selection, which considers photometric constraints. The photometric path is computed using $\bar{P}$ as described in Eq. (12). Note that the path from optimal selection avoids high frequency photometric change and is smoother than either the original or uniform selected paths. Shown on top, We can also visualize the frames being selected using uniform and optimal selection. The optimally selected frames are more photometrically consistent.*

However, we notice that directly applying the weighted filters to the transformation matrices results in color artifacts. These color artifacts usually are caused by different components of transformations being filtered independently and thus asynchronously; for example, the 6 independent variables in affine transformations should not be filtered independently. Wang *et al.* [WTL*14] tried to address this problem by smoothing in the PCA-encoded transformation space, but PCA decomposition and their use of only one principal component is limited to short video clips without much content change.

We address this issue by applying the weighted filter W on pixel values instead of on transformation parameters, and then recompute the desired transformation from the filtered pixel values. Specifically, we take the correspondence points p_i from the target frame i, transform its distribution to match each of its neighbor frames' distributions, and apply the weighted filter to get the desired color distribution as:

$$\hat{p}_i = \sum_{j=i-M/2}^{j=i+M/2} W(i,j)\left(T_{i,j}p_i\right). \tag{10}$$

These weighted color values represent the desired smoothly varying pixel values. We then compute a single transformation that aligns the original pixel values, p_i to these weighted pixel values as:

$$\arg\min_{\theta} \|\hat{T}(\theta)_i P_i - \hat{P}_i\| \tag{11}$$

where $\hat{P}_i$ is the desired distribution calculated in the previous step. $\hat{T}(\theta)$ is then applied to the entire frame to get $i^{'} = \hat{T}(\theta)i$, where i' is the corrected frame i. Correcting the video frames via this two-step process leads to results are more robust and artifact-free.

4. Photometrically Stable Frame Sampling

When generating fast-forward videos, the choice of frames can lead to different stabilized videos. Instead of applying uniform frame selection, we should select frames based on photometric constraints which can help skip degenerate frames such as over-exposed or under-exposed ones, and produce better stabilization. Our frame-sampling technique is similar to Joshi [JKT*15]; we define a novel binary photometric and unary blurriness cost and compute the optimal frame sampling by using dynamic programming while maintaining a user-specified frame sampling. The novel photometric costs we introduce are:

Photometric Cost: During frame sampling we want to remove degenerated frames that have large color changes and information lost (e.g. highly saturated or darkened). We define a simple photometric cost that characterizes the global image temperature and brightness:

$$C_p(i,j) = \|\bar{P}_i - \bar{P}_j\| \tag{12}$$

where the $\bar{P}_i$ and $\bar{P}_j$ denotes the mean correspondence value in frame i and j, in the *YCbCr* color space.

Blurriness Cost: To quantify the blurriness of each frame, we apply a Laplacian kernel to feature patches of each frame and compute the summed variance within the patches. The blurriness cost penalizes frames that contain large motion blur, either from camera motion or dynamic objects in the scene.

These costs can be added to the geometric costs introduced by [JKT*15] and used to sample optimal frames from a video sequence. Fig. 5 shows how sampling using our photometric costs can lead to more desirable stabilized videos.

5. Experiments

Synthetic videos To create the synthetic data, we first captured carefully controlled videos with a Canon 7D DSLR camera with locked manual camera settings. Therefore the original frames do not have any photometric variation introduced by camera settings, and brightness and color changes are caused by scene and illumination changes. We then applied high-frequency tone and color transformations to these videos. Specifically, high frequency jitter is created in the following two ways: 1) apply randomized color transform using our color model described in Sec. 3.1 2) manipulate individual frames in existing video editing software to insert temporal inconsistency. These altered video frames were then used to evaluate a variety of stabilization methods (see results on our self-captured video Fig. 6 and a publicly available video (photometrically stabilized) Fig. 7). It is possible to quantitatively evaluate on synthetic experiments given ground truth videos that are photometrically stable (for real videos there is no well-defined ground truth and thus we did not perform quantitative evaluation). We compute the RMSE for each frame with the ground truth video, and the results are shown in Fig. 8.

Real videos We also collected video datasets and applied our frame sampling (described in Sec. 4) to generate 16× fast-forward videos. These videos include an outdoor hiking sequence (Fig. 9) which contains both smooth scene illumination dynamics due to spatial changes in location, and high frequency tone jitter due to camera auto-adjustments (see f_1 in the top row), and another challenging video that transitions from outdoor to indoor (see Fig. 11).

Comparisons We compare our method with that from Farbman *et al*. [FL11] using code released by the original authors. For each of our videos, we manually choose an anchor frame that is of good image quality. While the authors claim that multiple anchor frames can be used, selecting these frames is a tedious manual task that is not practical for real videos. Fig. 6 and Fig. 7 compare the two methods on synthetic video sequences and Fig. 9 and Fig. 11 on two real videos. Across all these comparisons, our results remove high-frequency jitter while naturally adapting to changes in content and scene illumination. In contrast, Farbman *et al*.. try to match the appearance in all the frames leading to unnatural looking results and significant image artifacts (especially due to errors in the dense pixel correspondence that they rely on).

We tried to compare with Wang *et al*. [WTL*14] using the authors' code, but their method failed during the registration step since they assume neighboring frames to have little content variation, and their method is based on the quality of registration, while few correspondences are available across frames in our test videos. We also experimented with a global affine smoothing method, which is a factorization-based algorithm we implemented. Instead of factorizing the transformation matrices using *PCA* as in the work of Wang *et al*. [WTL*14], we decompose each affine transformation into 4 components: rotation, translation, shear and scale to avoid asynchronously smoothing different matrix entries. We then apply $L1$ smoothing to calculate a smooth path for each of the 4 components, warp the original value to its smooth path and reconstruct an affine transformation from the 4 warped components. This can be thought of as a version of [WTL*14] with our robust motion estimation and pair-wise transformation computation. However, we found the parameters of global smoothing hard to control, and the smoothed transformations introduce color artifacts (see Fig. 6, bottom).

Fig. 7 shows the result on synthetic outdoor scene. In this dataset, distant background scene does not change a lot, but moving crowd makes local correspondence matching unreliable. Farbman *et al*. [FL11] suffers from local artifact and incorrect adjustment due to inaccurate local correspondences, while our photometric stabilization effectively attenuates fluctuations in color and brightness.

Extension to time-lapse videos Our technique can also be applied to stabilize timelapse videos with large time span. Timelapse videos are usually captured from a static camera, but it may also have high frequency photometric caused by sudden illumination changes. In the example shown in Fig. 10, the brightness fluctuation comes from clouds in the sky getting in and out of the frame. Our photometric stabilization can distinguish high frequency jitter from gradual illumination changes and produce a result the jitter-free timelapse videos.

Running time Given a 100-frame speed-up video (i.e., 8× speedup from a 800-frame video) of resolution 1080×1920 (captured by an iPhone 6), our MATLAB single-thread implementation takes $212s$ to stabilize the entire sequence with 60% of the time spent on correspondence matching. However, because our technique relies on local temporal processing, the pair-wise correspondences and transformations can computed in parallel leading to significant time gain. In comparison, the method from Farbman *et al*. [FL11] takes $1957s$ to process the same test sequence with one anchor frame.

5.1. Conclusions

In this paper, we have presented a photometric stabilization method for fast-forward videos. Given a video input with a desired speedup factor, we perform photometrically optimal frame selection, and then apply stabilization on the selected frames to remove high frequency color and brightness fluctuations. Our technique is able to automatically detect and correct outlier frames and can also handle large content variations across frames without any prior information or manual anchor frame selection. The algorithm is designed to be computationally efficient and has the potential to be implemented extremely fast for real-time applications.

We evaluate our stabilization algorithm on both synthetic and real fast-forward videos with high frequency brightness and color fluctuations. We sample the original video frames using our optimal frame selection technique Sec. 4. We request the reviewers to see the supplementary video to evaluate the quality of our results. The supplementary material also contains more results and comparisons.

We focus on removing high-frequency color/tone variations. As

Figure 6: *Synthetic experiment on indoor scene dataset. (ROW 1) input sequence with high frequency photometric jitters (e.g.the f_2 frame contains sharp brightness change, and the f_{82} frame has incorrect white-balance; (ROW 2) stabilized frames using our proposed technique; (ROW 3) stabilized frames using Farbman et al. [FL11], in which they try to match the color of all frames to the select anchor frame. Also, local artifacts can be seen in the output due to up-sampling of their adjustment maps; (ROW 4) result of the global affine smoothing method; global smoothing cannot compensate the jitter of f_{81}, and can introduce color artifacts as seen in f_1.*

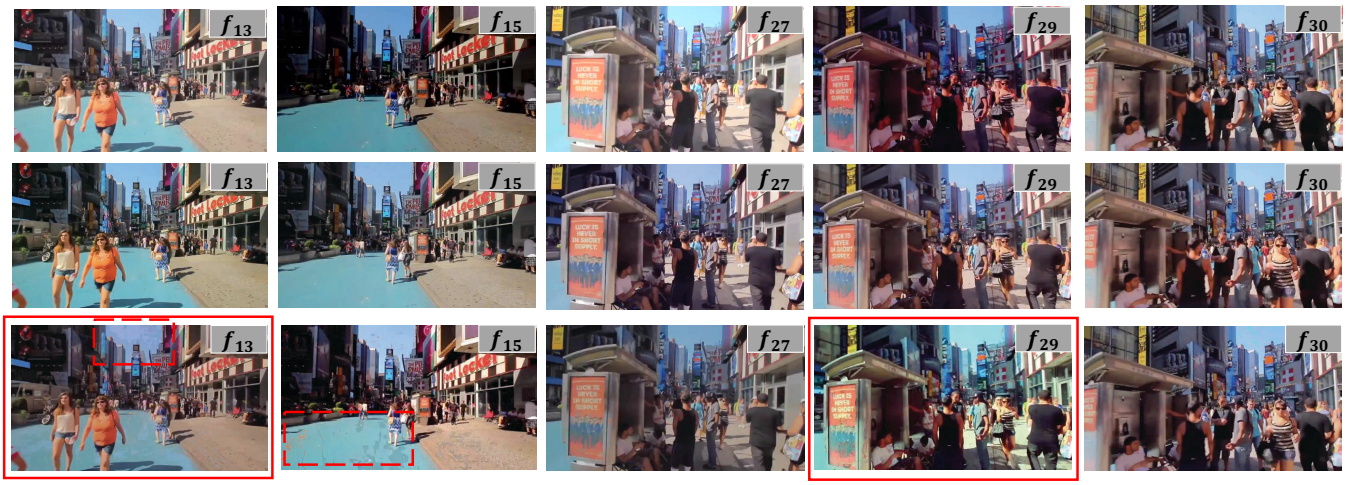

Figure 7: *Synthetic experiment on outdoor scene dataset. (ROW 1) input sequence with brightness and color photometric instability (e.g.the 2^{nd} and 3^{rd} contain brightness jitter, and 4^{th} frame contains color fluctuation); (ROW 2) stabilized frames using our stabilization method; (ROW 3) stabilized frames using Farbman et al. [FL11].*

a result low-frequency camera variations will not be corrected by this technique. We would like to address this in the future. We also would like to explore extensions of this framework to address the problem of photometrically aligning multiple video sequences of the same scene.

References

[BSPP13] BONNEEL N., SUNKAVALLI K., PARIS S., PFISTER H.: Example-based video color grading. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 39. 2

[BTS*15] BONNEEL N., TOMPKIN J., SUNKAVALLI K., SUN D., PARIS S., PFISTER H.: Blind video temporal consistency. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 196. 2

[FL11] FARBMAN Z., LISCHINSKI D.: Tonal stabilization of video. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 89. 2, 6, 7, 8, 9

[FSDH15] FRIGO O., SABATER N., DELON J., HELLIER P.: Motion

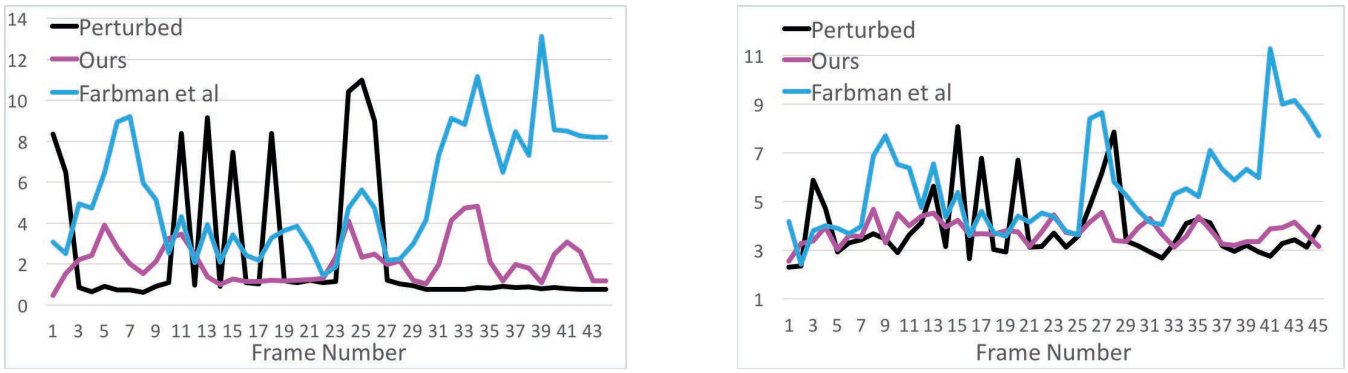

Figure 8: *Quantitative evaluations on synthetic video experiments. We compute the RMSE for each frame, between the stabilized video and the ground truth video, and compared with Farbman et al. [FL11]. (LEFT) result on synthetic indoor scene, corresponds to Fig. 6 (RIGHT) result on synthetic outdoor scene, corresponds to Fig. 7.*

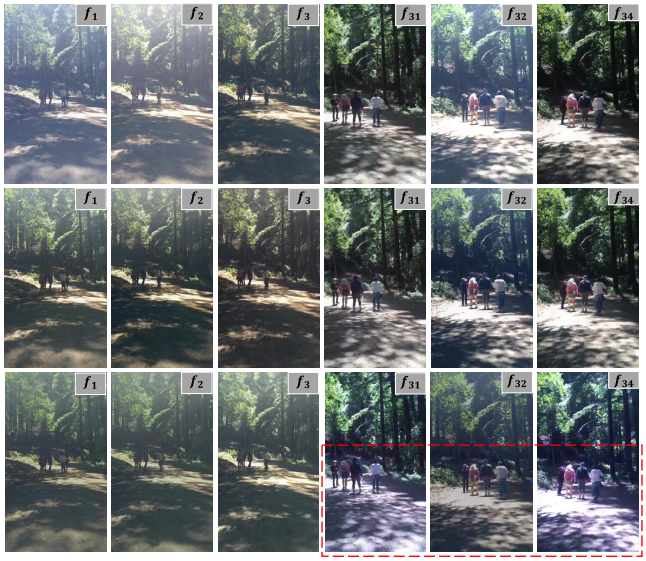

Figure 9: *Experiment on a real video dataset. (TOP) input frame sequence; (MIDDLE) corrected frames using our stabilization method; (BOTTOM) corrected frames using Farbman et al. [FL11]. In the input frame sequence, f_1 contains tone jitter, and f_1, f_2, f_{32} contain high frequency brightness fluctuation; the method from Farbman et al. [FL11] does not take into account the scene illumination change, and also introduces artifacts for frames such as f_{31} and f_{34}.*

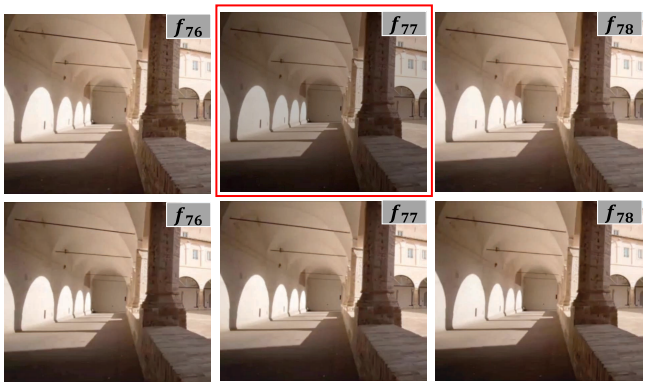

Figure 10: *Test on timelapse videos. (TOP) original frame sequence (BOTTOM) corrected frames using our proposed method.*

driven tonal stabilization. In *Image Processing (ICIP), 2015 IEEE International Conference on* (2015), IEEE, pp. 3372–3376. 2

[GKE11] GRUNDMANN M., KWATRA V., ESSA I.: Auto-directed video stabilization with robust l1 optimal camera paths. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 225–232. 2

[JKT*15] JOSHI N., KIENZLE W., TOELLE M., UYTTENDAELE M., COHEN M. F.: Real-time hyperlapse creation via optimal frame selection. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 63. 1, 5, 6

[KLL*12] KIM S. J., LIN H. T., LU Z., SÜSSTRUNK S., LIN S., BROWN M. S.: A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 12 (2012), 2289–2302. 2

[PHAP15] POLEG Y., HALPERIN T., ARORA C., PELEG S.: Egosampling: Fast-forward and stereo for egocentric videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 4768–4776. 1

[PKD05] PITIE F., KOKARAM A. C., DAHYOT R.: N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* (2005), vol. 2, IEEE, pp. 1434–1439. 2

[PKD07] PITIÉ F., KOKARAM A. C., DAHYOT R.: Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding 107*, 1 (2007), 123–137. 2

[RAGS01] REINHARD E., ASHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. 2

[RRKB11] RUBLEE E., RABAUD V., KONOLIGE K., BRADSKI G.: Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision* (2011), IEEE, pp. 2564–2571. 2

[RTG00] RUBNER Y., TOMASI C., GUIBAS L. J.: The earth mover's distance as a metric for image retrieval. *International journal of computer vision 40*, 2 (2000), 99–121. 3

Figure 11: *Experiment on a real video dataset. (TOP) input frame sequence; (MIDDLE) corrected frames using our method; (BOTTOM) corrected frames using Farbman et al. [FL11]. The input frame sequence contain brightness fluctuation due to the change from outdoor to indoor. Farbman et al. [FL11] fails to handle the over-exposed frame like* f_{23}.

[VCB14] VAZQUEZ-CORRAL J., BERTALMÍO M.: Color stabilization along time and across shots of the same scene, for one or several cameras of unknown specifications. *IEEE Transactions on Image Processing 23*, 10 (2014), 4564–4575. 2

[VCB15] VAZQUEZ-CORRAL J., BERTALMÍO M.: Simultaneous blind gamma estimation. *IEEE Signal Processing Letters 22*, 9 (2015), 1316–1320. 3

[WTL*14] WANG Y., TAO D., LI X., SONG M., BU J., TAN P.: Video tonal stabilization via color states smoothing. *IEEE transactions on image processing 23*, 11 (2014), 4838–4849. 2, 5, 6