https://doi.org/10.2352/ISSN.2470-1173.2019.9.IMSE-373

Auto White Balance Stabilization in Digital Video

Niloufar Pourian, Rastislav Lukac; Intel Corporation, Santa Clara, CA, USA

Abstract

Auto white balance (AWB) plays a key role in a digital camera system, and determines image quality to a large extent. Existing AWB algorithms are not completely reliable and often produce inconsistent temporal estimates for consecutive frames, resulting in abrupt color changes in the output video stream. This paper presents an efficient approach to stabilize AWB estimates in a video by detecting scene and/or light source changes and adaptively setting appropriate convergence times.

Introduction

Automatic white balancing (AWB) mimics the chromatic adaptation functionality of the human visual system [6]. It refers to the process of correcting the image for color shifts attributed to the color of a light source. Thus, AWB methods have an important role in color imaging devices, such as digital cameras, aiming at a faithful representation of the visual scene in real time, without any manual adjustment from the user. In video or live-view (preview) modes, AWB estimates are usually produced for each frame of a continuous video stream. Therefore, stabilizing AWB estimates is crucial to produce visually-pleasing video.

Stabilizing AWB is usually approached by calculating the final AWB estimate for the current frame as a weighted average of temporal AWB estimates obtained for the current frame and several prior frames. The number of prior frames to be considered is typically defined relative to a constant convergence time assigned to AWB algorithm during tuning phase. AWB algorithms are not completely reliable and often produce different or even incorrect temporal estimates for consecutive frames, resulting in abrupt color changes in the output video stream. One solution is to increase the convergence time. However, having a single large convergence time has a drawback of producing a lag when a fast AWB response is desired, for instance, in situations with the light source and/or the scene changes.

There is a wealth of published literature on AWB algorithms for digital photography emphasizing the importance of the accurate estimate of white point on image quality [1, 2, 3, 4, 5, 7, 8, 9, 10]. [4] allows for a white balance control with reference to a white portion of an object in the field of view of the video camera. Similarly, [3] proposes an approach to a local automatic white balance algorithm. Also, [5] proposes an algorithm that finds more proper white pixels to calculate the averaged chromatic aberration and improve the precision of the estimated color temperature. In addition, the work in [2] proposes an approach to compensate the color differences by deciding whether the color distortion of the scene is caused by a light source or a chromatic object. However, the aforementioned works mainly focus on accurate AWB computation for any given frame independently. While [7] is based on a multiframe approach for white balancing of digital color images, their main contribution is on modifying the exposure time independently for each color component.

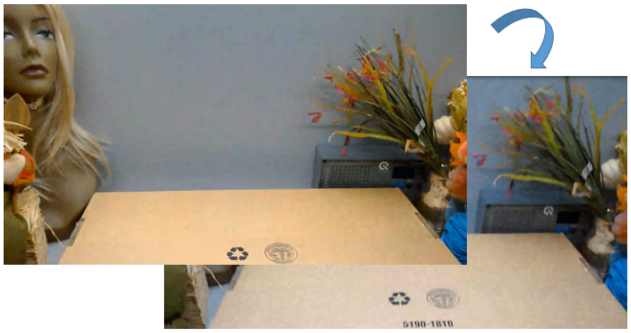


Figure 1. Example of the dramatic change in color due to inconsistent AWB results. Frames shown are within a small window in a video w/o abrupt changes in light source and/or scene. Best viewed in color.

In this work, we propose a method on stabilizing AWB estimates for continuous video stream, while providing the flexibility of different convergence rates under various scene/light source changes for the end user.

Approach

The proposed approach adaptively sets appropriate convergence times by detecting scene and light source changes through analysis of temporal changes between video frames using several feature signals, such as the correlated color temperature, color statistics, and high-frequency image information. The proposed method is effective, computationally efficient, and produces visually pleasing video.

Algorithm Description

This work proposes a method that adaptively sets appropriate convergence times by detecting scene and/or light source changes. In particular, the proposed method aims at recognizing the following scenarios:

- Scene change and no light source change
- Light source change and no scene change
- Scene change and light source change

Each of the above scenarios is associated with a specific convergence time, with the largest being used in situations when no light source and no scene change are detected, and the smallest being used in situations with both light source and scene changes. This is achieved by analyzing temporal changes in several feature signals, obtained for each frame of the input video stream. These feature signals include the correlated color temperature (CCT) values, color statistics, and high-frequency image information.

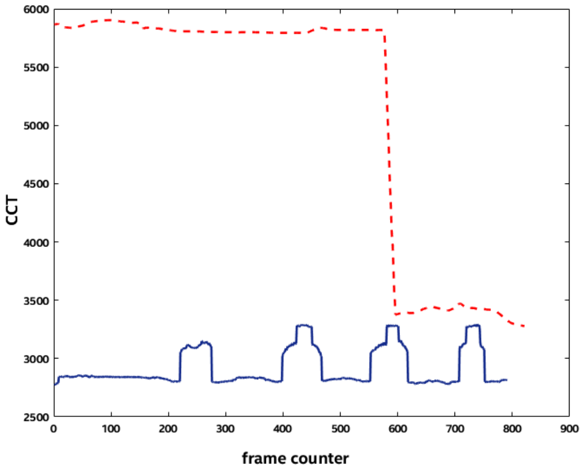


Figure 2. *Illustration of CCT values associated with a set of continuous frames of a hallway carpet with negligible change in light source (blue), and a video of a lab booth undergoing a light source change (dotted red). Best viewed in color.*

The CCT value gives a measure of light source color appearance defined by the proximity of the light sources chromaticity coordinates to the blackbody locus. This value can be estimated by finding the closest point to the light source's white point on the Planckian locus. Since most illuminants can be characterized by the color temperature of the light source and various 3A (auto-exposure, auto white-balance, and auto-focus (AF)) systems usually output some form of CCT information, analyzing temporal differences in CCT values gives a useful information on potential light source changes. Therefore, the proposed method aims at finding the frames with large differences between their estimated CCT values. In our experiments, the change in CCT values is computed using the absolute difference between the CCT values associated with frame i and frame j. This is summarized as following:

$$\Delta CCT(i,j) = |CCT^i - CCT^j| \tag{1}$$

where CCT^i denotes the CCT value at frame i. In addition, Figure 2 demonstrates how CCT values could be a useful indicator on light source change by illustrating sample CCT values associated with contiguous frames of a video undergoing a negligible light source change versus a dramatic light source change. It can be seen that the maximum ΔCCT between continuous frames is about 300 for the video with negligible light source change and is about 2500 for the video with drastic light source change.

The proposed method also relies on color statistics which are used to obtain both global and localized information on each frame. In the former case, one histogram per color channel is calculated to collect the statistics from the entire frame. In the latter case, localized color histograms are obtained for various spatial partitions of each frame. For instance, the frame can be divided into halves (top and bottom, right and left), quadrants, and some other non-overlapping or overlapping blocks based on some predetermined criteria. Histograms can be calculated in RGB or some other suitable color space, such as YUV, LAB, etc. Obtained global and localized histograms are combined and normalized for each color channel. In order to combine the global and localized color channel information associated with each frame, one can either concatenate these histograms, or compute a weighted average of them. Normalization can also be applied using different techniques, e.g. ℓ_1, ℓ_2, and ℓ_∞. After completing the normalization, each frame is represented by a vector that combines normalized histograms from all color channels. The vectors obtained for different frames are analyzed for temporal changes. The difference between these vectors can be computed using Hellinger distance, Euclidean distance, Minkowski distance, or inner product. In our experiments, the color statistics are created by concatenating the ℓ_1 normalized global and local histograms associated to each color channel R, G, and B, followed by another ℓ_1 normalization. In this paper, we refer to these histograms as spatial pyramid of color histograms or **in short color statistics** (*CS*) [13]. Furthermore, we use Hellinger distance to measure the differences between color statistics of frames i and j:

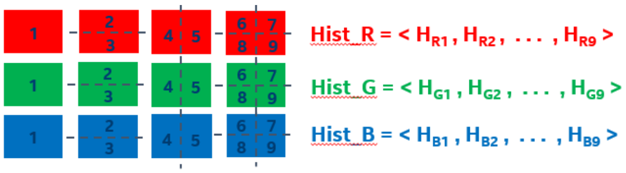

1
2
3
4
5
6
7
8
9
Hist_R = < H_R1 , H_R2 , . . . , H_R9 >
Hist_G = < H_G1 , H_G2 , . . . , H_G9 >
Hist_B = < H_B1 , H_B2 , . . . , H_B9 >

Figure 3. *Illustration of algorithm based on spatial pyramid of color histograms. Best viewed in color.*

$$\Delta CS(i,j) = \sqrt{\sum_{k=0}^{K} (h_k^{(i)} - h_k^{(j)})^2} \tag{2}$$

with $h^{(i)}$ being the normalized spatial pyramid of color histogram associated to frame i, and K denoting the total number of bins in each histogram.

The final feature signal caries the high-frequency information associated with each frame, putting a particular emphasis on edges and corners. As before, the feature signal extracted for different frames is analyzed for temporal changes. The high-frequency information can be extracted using feature detectors, such as Scale-Invariant-Feature-Transforms (SIFT) [11] or Speeded-Up-Robust-Features (SURF) [12], and the degree of dissimilarity between the frames is measured by the inverse of the number of matching points of interest. In addition, the feature signal can be obtained through high-pass filtering of individual frames, for instance, using the Sobel, Canny, Laplacian or some other edge operator. This is followed by evaluating the temporal differences in high-frequency contents using Manhattan distance, but other distance metrics such as Euclidean distance can be used as well. This process can be applied to the gray scale version of the RGB image or the luminance channel when the captured image is transformed to YUV, YCbCr, LAB, LUV, HSV, or some other suitable color space which separates the luminance from other color attributes, such as chrominance, hue, and saturation. It is also possible to process each color channel separately and then combine the intermediate results. Alternatively, the high-frequency image information can be extracted in a way that takes advantage of color correlations; for instance, using vector edge operators. In this paper, the high frequency information is computed by high-pass filtering the individual frames in gray scale

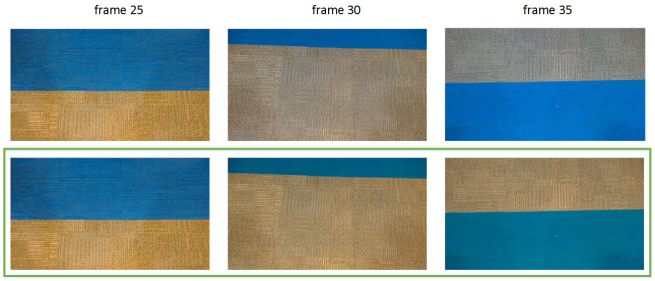


Figure 4. *Video with negligible scene and light source changes. It shows a carpet taken in a hallway under stable illumination. From left to right, displayed images correspond to frames 25, 30, and 35, respectively. (Top) Using shorter AWB convergence times results in significant color changes. (Bottom) Using longer AWB convergence times keeps color information more uniform between the frames. The proposed method correctly recognizes that neither scene change nor light source change occurred and thus adapts longer convergence times. Best viewed in color.*

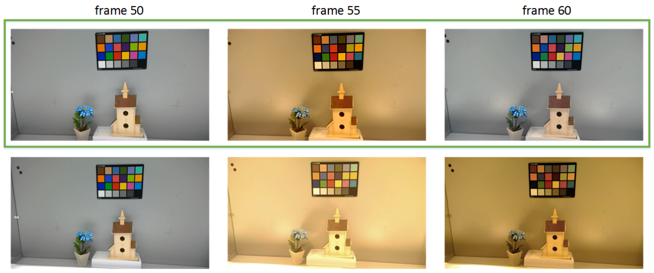


Figure 5. *Video with significant light source changes. This video is captured in the lab booth by enforcing sudden light source changes. From left to right, displayed images correspond to frames 50, 55, and 60, respectively. (Bottom) Using longer AWB convergence time shows a slow response to illumination changes. (Top) Using shorter AWB convergence time produces desired results. The proposed method correctly detects a light source change and thus adapts shorter convergence times. Best viewed in color.*

using Sobel operator and measuring the ℓ_1 distance between the edge vectors of consecutive frames.

The proposed method in its entity or any of its design elements (i.e., temporal change detection based on CCT, color statistics, and high-frequency image information), can be applied to full-resolution video frames or their subsampled (downsized) or otherwise processed (e.g., noise reduction) versions. Regardless of such pre-processing, there are several ways how the proposed method can be applied in practice. For instance, one can choose to evaluate the differences between the two consecutive frames, or define a fixed step function to select the frames that are compared with each other. Additionally, one can select an adaptive step to evaluate the differences between the current frame and the last frame with a detected scene and/or light source change. In any of these processing modes, a large difference between CCT values for two considered frames is interpreted as a light source change. Smaller CCT differences can also suggest a light source change, but only if color histograms for two considered frames vary significantly while the associated high-frequency contents remain similar. If neither of these two cases occurs, the proposed method checks for a possible scene change; this situation is usually associated with large differences in both color histograms and high-frequency content. In all other cases, changes between the two frames are considered negligible.

Evaluation

The actual evaluation of whether the differences in CCT values, color statistics, and high-frequency image information between the two frames are large enough to indicate a light source change and/or a scene change is done using tunable parameters. If any of these events occurs, the initial AWB convergence time can be replaced with some shorter value. In one example, the shortest convergence time is used when a light source change is detected, while some longer convergence time is used when a scene change is detected. In another example, the shortest convergence time is used when the CCT difference exceeds a predefined high CCT threshold, some longer convergence time is used when the CCT difference exceeds a predefined low CCT threshold and simultaneously the color statistic (CS) difference exceeds a predefined CS threshold while the high-frequency content (HFC) difference is smaller than a predefined HFC threshold, and even longer convergence time, but still smaller than the initial AWB convergence time, is used when the CS difference exceeds the CS threshold and simultaneously the HFC difference exceeds the HFC threshold. In yet another example, the final convergence time is determined using a function of one or more of the CCT, CS and HFC differences. For instance, the final convergence time can be adaptively calculated as a combination (i.e., weighted average) of the two predefined convergence times, where the weights can be inversely proportional to the difference values. Alternatively, the weights can be calculated using the exponential function or some other suitable function.

Since our goal is to implement the proposed method in an efficient and scalable way on various target platforms, the final design choices take advantage of available resources and computationally simple approaches as much as possible. Namely, the proposed method reuses both the low-resolution frame color statistics and the CCT values that are commonly used/produced by camera control algorithms, instead of calculating the CCT values, histograms, and high-frequency contents from the full-resolution frames. It is worth noting that to obtain global and localized color histograms, nine histograms with eight bins are defined for each of the RGB color channels. These histograms correspond to whole frame and its top half, bottom half, left half, right half, and four quadrants, respectively. This is illustrated in Figure 3. The nine histograms associated with each color channel are concatenated and ℓ_1 normalized. The three normalized histograms (one per color channel) are then further concatenated, resulting in a feature vector with 216 elements. The differences between the normalized feature vectors obtained for different frames are evaluated using the Hellinger distance. To calculate the high-frequency information associated with each frame, the luminance channel (obtained from the Bayer color filter array data) of low-resolution frames is subject to high-pass filtering. The degree of dissimilarity in high-frequency contents between the two frames is computed using Manhattan distance. The final convergence time is set to its shortest allowed value when the absolute CCT difference exceeds a predefined high CCT threshold, while in any other situation the final convergence time is equivalent to a weighted combination of the shortest and longest allowed convergence times. The weight associated with the shortest allowed convergence time is set as the difference between the maximum possible HFC difference value and the actual HFC difference value scaled by a predetermined HFC factor. The weight associated with the longest allowed convergence time is obtained by subtracting the other weight (for the shortest convergence time) from one.

Experimental Results

The effectiveness of the proposed method was tested with different scene complexity and illumination. Here, we show the recording of a hallway carpet under stable illumination (Figure 4) as well as the recording of a laboratory scene with some abrupt illumination changes (Figure 5). More specifically, Figure 4-a shows how small camera motion can result in abrupt color changes caused by the use of short convergence times in AWB, even when both the scene and illumination remain practically constant. As demonstrated in Figure 4-b, this issue can be alleviated by choosing larger convergence. On the other hand, Figure 5-a shows how long convergence times have a drawback of producing a lag, when a fast AWB response is desired in the situations with a light source change. In this case, acceptable results can be produced using shorter convergence times, as illustrated in Figure 5-b. As suggested by these examples, the proposed method adjusts the AWB convergence time in each scenario and produces high-quality results by setting longer convergence times when no significant change is detected and adapting shorter convergence times when a change is detected.

Discussion and Conclusion

This work presents a robust and computationally efficient method for dealing with inconsistencies in AWB estimates associated with individual frames of live-view (preview) or capture video streams. We proposed a method on stabilizing AWB estimates and providing the flexibility of different convergence rates under various scene/light source changes for the end user. The proposed work can benefit any digital camera, cell-phone, tablet, and ISP manufacturer since digital video preview and video capture are must-to-have features in such products, and producing stable and consistent AWB estimates in digital video is crucial for good user experience.

Acknowledgments

The authors would like to thank Gary Sun and Changyeon Jo for their helpful discussions.

References

[1] Koji Takahashi, Method and apparatus for correcting white balance, method for correcting density and recording medium on which program for carrying out the methods is recorded, US Patent 7, 146, 041. (2006).

[2] Yoon Kim, et.al., A video camera system with enhanced zoom tracking and auto white balance, IEEE Transactions on Consumer Electronics, pg. 428. (2002).

[3] Jun-yan Huo, et.al., Robust automatic white balance algorithm using gray color points in images, IEEE Transactions on Consumer Electronics, pg. 541. (2006).

[4] Toshiharu Kondo, et.al., Digital color video camera with auto-focus, auto-exposure and auto-white balance, US Patent 5, 093, 716. (1992).

[5] Zhou Rongzheng He Jie Hong Zhiliang, Adaptive Algorithm of Auto White Balance for Digital Camera, Journal of Computer Aided Design & Computer Graphics, pg. 24. (2005).

[6] Edmund Y Lam, et.al., Automatic white balancing in digital photography, Single-sensor imaging: Methods and applications for digital cameras, pg. 267. (2008).

[7] Radu Ciprian Bilcu, Multiframe auto white balance, IEEE Signal Processing Letters, pg. 165. (2011).

[8] Wonwoo Jang, et.al., Auto white balance system using adaptive color samples for mobile devices, IEEE Asia Pacific Conference on Circuits and Systems, pg. 1462. (2008).

[9] Yasushi Takagi, et.al., White balance adjusting device for video camera, US Patent 5,270,802. (1993).

[10] Toshiki Miyano, Auto white balance apparatus, US Patent 6,727,942. (2004).

[11] David G Lowe, Object recognition from local scale-invariant features, The proceedings of the seventh IEEE international conference on Computer vision, pg. 1150. (1999).

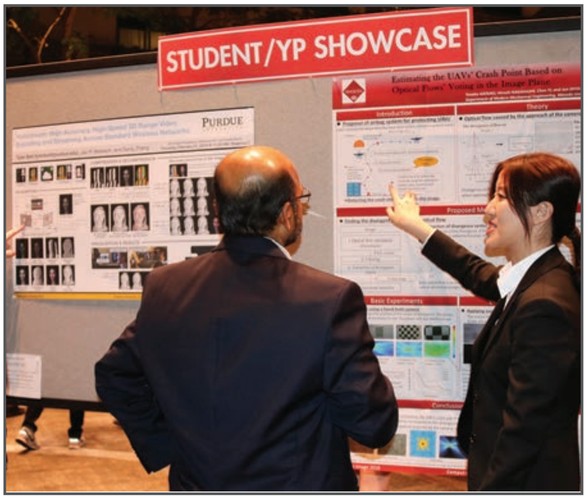
[12] Herbert Bay, et.al., Surf: Speeded up robust features, European conference on computer vision, pg. 404. (2006).

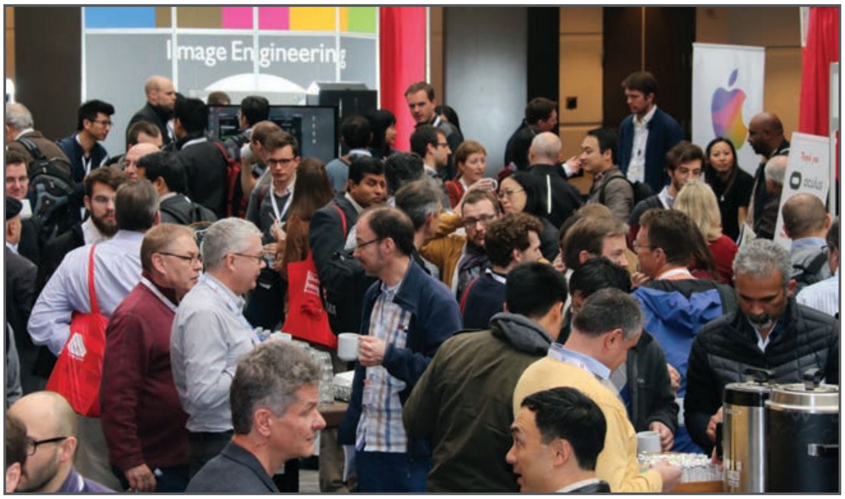
[13] Svetlana Lazebnik, et.al., Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pg. 2169. (2006).

Author Biography

Niloufar Pourian received her BS, MS, and PhD in Electrical and Computer Engineering from University of California at Santa Barbara in 2010, 2013, and 2015, respectively. She is currently a research scientist at Intel Labs. Her research is mainly focused on computer vision, image processing, and computational photography. Dr. Pourian is the recipient of the UC Regents Fellowship and Deans Doctoral Scholar Award in 2007 and 2010, respectively. She is also the recipient of the Intel Division Recognition Award in 2018.

Rastislav Lukac received M.S. (Ing.) and Ph.D. degrees in telecommunications from the Technical University of Kosice, Slovak Republic, in 1998 and 2001, respectively. He is currently a technical lead and architect in Intel camera imaging team. Dr. Lukac is the author of five books and four textbooks, a contributor to twelve books and three textbooks, and he has published more than 200 scholarly research papers in the areas of digital camera image processing, color image and video processing, multimedia security, and microarray image processing. He holds about 50 patents in the areas of digital color imaging and pattern recognition. He is the recipient of the 2003 North Atlantic Treaty Organization / National Sciences and Engineering Research Council of Canada (NATO/NSERC) Science Award, the Most Cited Paper Award for the Journal of Visual Communication and Image Representation for the years 2005−2007, the 2010 Best Associate Editor Award of the IEEE Transactions on Circuits and Systems for Video Technology, and the author of the #1 article in the ScienceDirect Top 25 Hottest Articles in Signal Processing for April−June 2008.





- AR is a fundamental and transformative interface and platform.
- 20+ years from my initial working demo to today's AR displays
- Optimistic that it won't take another 20 to reach ubiquitous consumer AR!