Greyworld White Balancing with Low Computation Cost for On-Board Video Capturing

Peng Wu and Yuxin (Zoe) Liu; Hewlett-Packard Laboratories, Hewlett-Packard Co.; Palo Alto, CA, 94304, USA

Abstract

White balancing is a process commonly used in digital photography to estimate the illuminant and then compensate the color appearance so that the white color appears as white regardless of the variety in the illumination condition when the picture was captured. In this paper, we propose a low computation cost approach to realize white balancing for on-board digital video capturing. The Greyworld approach has been widely adopted and considered as the basis for many automatic white balancing algorithms. When the Greyworld approach is applied to digital videos, each video frame in the sequence is treated independently. The two modules in the approach, namely the multiplier estimation and the frame modification, are repeated on every frame without taking into account the temporal correlation across adjacent frames. In fact, when a segment of a video clip is captured under a fixed lighting condition and if the objects in the scene do not move sufficiently rapidly, the multipliers for different frames often possess strong correlation. Such an observation leads to the following two ideas to reduce the overall computation cost for white balancing on-board videos: 1) Detecting and keeping track of the illuminant change across frames to determine if the multiplier estimation for a new frame is necessary; 2) Instead of using the multipliers to modify one entire frame, a block-based approach is proposed to compare blocks of the current frame with the co-located ones in the previous frame and to perform color modification on selected blocks. Using our proposed approach the computational complexity is reduced while the performance is maintained. As a trade-off we need to maintain the frame buffer as well as the multiplier buffer.

Introduction

White balancing is a process commonly used in digital photography to estimate the illuminant and then compensate the color appearance so that the white color appears as white regardless of the variety in the illumination condition when the picture was captured. In this paper, we propose a low computation cost approach to realize white balancing for on-board digital video capturing.

The Greyworld approach has been widely adopted and considered as the basis for many automatic white balancing algorithms. Most (consumer) videos may endure a variety of illuminations within a short time window. The Greyworld approach determines the chromaticity of the illuminant from the average of all the pixels in one image, assuming that the average color of the scene is gray and that any departure from this average in the image is caused by the illuminant.

When the Greyworld approach is applied to digital videos, each video frame in the sequence is treated independently. The two modules in the approach, namely the multiplier estimation and the frame modification, are repeated on every frame without taking into account the temporal correlation across adjacent frames. In fact, when a segment of a video clip is captured under a fixed lighting condition and if the objects in the scene do not move sufficiently rapidly, the multipliers for different frames often possess strong correlation. Such an observation leads to the following two ideas to reduce the overall computation cost for white balancing on-board videos:

(1) Detecting and keeping track of the illuminant change across frames to determine if the multiplier estimation for a new frame is needed;

(2) Instead of using the multipliers to modify one entire frame, a block-based approach is proposed to compare blocks of the current frame with those co-located ones in the previous frame; A subset of blocks in the current frame are then identified to represent the portion of the current frame that are different from the previous frame in terms of the illuminant; The color modification is then only performed on those identified blocks.

Using our proposed approach the computational complexity is reduced while the performance is maintained. Compared with other Greyworld-based white balancing approaches for video capturing, our proposed technique has the distinguished advantage on the computation cost. Through taking the temporal correlation across adjacent frames plus the consistency of illuminant into account, the proposed technique can lower the amount of multiplication operations involved in the color modification. As a trade-off we need to maintain the frame buffer as well as the multiplier buffer.

Greyworld-Based White Balance Technique for On-board Digital Video Capture

Greyworld Algorithm for White Balance

Various algorithms have been developed to estimate and adjust illuminations. Considering the computational complexity of the algorithm and the fact that (consumer) video capturing may have different illuminations within a single video clip, we mainly investigate the Greyworld approach in this paper. In the Greyworld approach, the chromaticity of the illuminant is determined from the average of all the pixels in one image. It assumes that the average color of the scene is gray and that any departure from this average in the image is caused by the illuminant. Specifically, given an image in the RGB space $I(x,y)$ and letting $I_{\mathrm{R}}(x,y)$, $I_{\mathrm{G}}(x,y)$ and $I_{\mathrm{B}}(x,y)$ represent the image pixels in the R, G, and B channels at position (x,y), the Greyworld algorithm searches for a transform in the color space as follows:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} \alpha R & & \\ & \beta G & \\ & & \gamma B \end{bmatrix} \tag{1}$$

that applies to the $I_{\mathrm{R}}(x,y)$, $I_{\mathrm{G}}(x,y)$ and $I_{\mathrm{B}}(x,y)$ components. The multipliers α, β, and γ are obtained as follows:

$$\alpha = \delta / E(I_{\mathrm{R}}(x,y)) , \tag{2.1}$$

$$\beta = \delta / E(I_{\mathrm{G}}(x,y)) , \tag{2.2}$$

$$\gamma = \delta / E(I_{B}(x,y)) , \tag{2.3}$$

where $E(I_{\mathrm{R}}(x,y))$, $E(I_{\mathrm{G}}(x,y))$ and $E(I_{\mathrm{B}}(x,y))$ are the mean of $I_{\mathrm{R}}(x,y)$, $I_{\mathrm{G}}(x,y)$ and $I_{\mathrm{B}}(x,y)$, respectively, and

$$\delta = \min\{E(I_R(x,y)), E(I_G(x,y)), E(I_B(x,y))\} . \tag{3}$$

The modified image using greayworld is then obtained as:

$$I_R^*(x,y) = \alpha I_R(x,y), \tag{4.1}$$

$$I_G^*(x,y) = \beta I_G(x,y), \tag{4.2}$$

$$I_B^*(x,y) = \gamma I_B(x,y). \tag{4.3}$$

Greyworld-Based White Balance for Video Capture

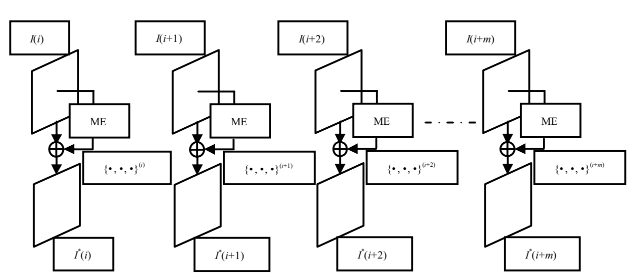

Figure 1 Fundamental procedure of Greyworld-based white balance for video capture

In [1] and [2], using the Greyworld-based white balance approach for still/video capture is addressed. As illustrated in Figure 1, in digital video capture, a video clip is the output of capturing a sequence of frames continuously within a certain time period. The white balance scheme described in [1] and [2] is essentially to apply the Greyworld approach, as introduced in the previous section, to each of the frame in the sequence to generate a modified frame sequence as the final captured video. More specifically, for the ith frame $I(i)$, the multiplier estimation (ME) block, shown in Figure 1, utilizes Eq. (2.1), (2.2) and (2.3) to compute a set of multipliers $P(i) = \{\alpha, \beta, \gamma\}^{(i)}$ and such a set of multipliers are used to modify the ith frame using Eq. (4.1), (4.2) and (4.3) to obtain the corrected frame $I^*(i)$.

A Low Computation Cost Greyworld Technique for On Board Video Capture

In the scheme described in Figure 1, each frame in the sequence is treated independently. The multiplier estimation and frame modification are repetitively conducted on each frame without considering the temporal correlation across adjacent frames. In fact, when a segment of video clip is captured under a fixed lighting condition and the objects in the scene do not move rapidly, the multipliers often possess strong correlation. Such an observation leads to the follow two ideas to reduce the computation cost:

(1) Detect the illumination change across frames to determine if the multiplier estimation is needed;

(2) Instead of using the multipliers to modify the entire frame, a block-based approach is proposed to compare blocks of the current frame with those co-located ones in the previous frame to identify a subset of blocks in the current frame; These identified blocks represent the portion of the current frame that is different from the previous frame in terms of the illumination condition. The color modification is then only performed on the identified blocks.

Illumination Change Detection

To detect the change of illumination across frames, the color histogram is widely used, as described in [3]. Given a frame $I(i)$, let $H_i(j)$ denote its color histogram, where $j = 1,2,\ldots,G$ and G is the total number of bins used to form the histogram. The distance between the histograms of two adjacent frames $I(i)$ and $I(i+1)$ is obtained as:

$$SD_i = \sum_{j=1}^{G} \left| H_i(j) - H_{i+1}(j) \right| . \tag{5}$$

If the difference SD_i is larger than a given threshold T, then an illumination change is declared. To set the threshold, it usually requires the calculation of the mean μ and the standard deviation σ of SD_i over $I(i)$, $i = 1,2,\ldots,m,\ldots$. The threshold can be obtained as:

$$T = \mu + \alpha\sigma . \tag{6}$$

A typical value for α is 5 or 6.

For the on-board video capture, it is not realistic to compute the mean μ and the standard deviation σ over the entire video clip to be captured, for that requires the buffering of all the captured frames. Instead, an adaptive illumination change detection algorithm has been proposed, in which only a small number of histograms need to be buffered. A similar approach has been suggested in [4] for the purpose of video summarization. Let HB denote the histogram buffer, which keeps K histograms. The adaptive illumination change detection algorithm is described as follows:

Step 1:	For frame $I(i)$, $i = 1,\ldots,K$, obtain H_i, and let $HB_{K-i} = H_i$;
Step 2:	Obtain $SD_l = \sum_{j=1}^{G} \left\| HB_l(j) - HB_{l+1}(j) \right\|$, $l = 0,\ldots,K-2$;
Step 3:	Obtain $\mu_{HB} = mean(SD_l)$, $\sigma_{HB} = std(SD_l)$, and $T_{HB} = \mu_{HB} + \alpha\sigma_{HB}$;
Step 4:	Set $i = K+1$;
Step 5:	Obtain H_i from $I(i)$;
Step 6:	Obtain $SD' = \sum_{j=1}^{G} \left\| HB_{K-1}(j) - H_i(j) \right\|$;
Step 7:	If $SD' > T_{HB}$, declare illumination change between $I(i)$ and $I(i-1)$ and go to Step 8, else go to Step 8;
Step 8:	Set $HB_l = HB_{l+1}$, $l = 0,\ldots,K-2$, an $HB_{K-1} = H_i$;
Step 9:	Set $SD_l = SD_{l+1}$, $l = 0,\ldots,K-3$, and $SD_{K-2} = SD'$;
Step 10:	Obtain $\mu_{HB} = mean(SD_l)$, $\sigma_{HB} = std(SD_l)$, and $T_{HB} = \mu_{HB} + \alpha\sigma_{HB}$;
Step 11:	Set $i = K+2$, go to Step 5.

Adopting the above algorithm, each input frame $I(i)$, $i = K+1,\ldots$, can be examined to determine whether it is at the boundary of an illumination change. To leverage the temporal correlation of the multipliers under the same or similar illumination, we only perform the multiplier estimation when the frame is determined as the boundary frame. More specifically, we maintain a multiplier buffer to keep track of the multipliers used for modifying the current frame. The values of multipliers will be updated through multiplier estimation only when the incoming frame is declared to be the boundary of luminance change. Otherwise, the buffered multipliers will be used to modify the incoming frame.

Block-Based Color Correction

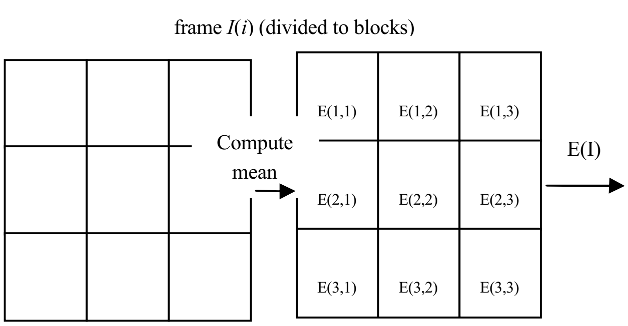

Figure 2 Illustration of block-based computation of means

If frames are identified as belonging to the same luminance condition, the same set of $P(i) = \{\alpha, \beta, \gamma\}^{(i)}$ is used throughout all these frames to perform the color modification as described in Eq. (4.1), (4.2) and (4.3). Further, we can leverage the temporal correlation of consecutive frames to reduce the computation cost. We first divide $I_R(i)$, $I_G(i)$ and $I_B(i)$ into blocks of equal size. Then, when computing the mean of $I_R(i)$, $I_G(i)$ and $I_B(i)$, not only $E(I_R(i))$, $E(I_G(i))$ and $E(I_B(i))$ are computed, but also the mean of each block of the $I_R(i)$, $I_G(i)$ and $I_B(i)$, as illustrated in Figure 2.

Suppose for $I_p(i)$, $p = R, G, B$, we divide it into $M \times N$ blocks. The mean computation will result a mean of the entire $I_p(i)$ and means of blocks $E_p(i,m,n)$, where $m = 1,\ldots,M$ and $n = 1,\ldots,N$. To perform block-based modification, we examine the mean $E_p(i,m,n)$ with $E_p(i-1,m,n)$ and the following procedure is conducted:

Procedure for block-based color modification:
Given $I_p(i-1)$, $I^*_p(i-1)$, $E_p(i-1,m,n)$, and $E_p(i,m,n)$,
for block (m,n) of $I_p(i)$, {
    if not ($\left|E_R(i,m,n) - E_R(i-1,m,n)\right| < T_E$ and
        $\left|E_G(i,m,n) - E_G(i-1,m,n)\right| < T_E$ and
        $\left|E_B(i,m,n) - E_B(i-1,m,n)\right| < T_E$)
    then {
        for all the pixels belonging to block $(x,y) \in block(m,n)$ of $I_p(i)$, compute block (m,n) of $I^*_p(i)$ as follows:
        $I^*_R(i,x,y) = \alpha^{(i)} I_R(i,x,y)$,
        $I^*_G(i,x,y) = \beta^{(i)} I_G(i,x,y)$,
        $I^*_B(i,x,y) = \gamma^{(i)} I_B(i,x,y)$;
    }
    else {
        for all the pixels belonging to block $(x,y) \in block(m,n)$ of $I_p(i)$, compute block (m,n) of $I^*_p(i)$ as follows:
        $I^*_R(i,x,y) = I^*_R(i-1,x,y)$,
        $I^*_G(i,x,y) = I^*_G(i-1,x,y)$,
        $I^*_B(i,x,y) = I^*_B(i-1,x,y)$.
    }
}

The idea behind this procedure is that if the block differences on *R*, *G*, and *B* channels across frames are all very small, we consider it as a strong indication that the two blocks located in (m,n) in frame $I_p(i)$ and $I_p(i-1)$ are very similar. Since we use the same set of multipliers, we can safely set the block (m,n) in frame $I^*_p(i)$ to be the same as that of $I^*_p(i-1)$. If the block differences are not all very small, we just perform the block-based modification using the buffered multipliers. In this way, for blocks that have consistent pixel values across consecutive frames, we reduce the computation by skipping the pixel modification that involves the multiplication operation. As a trade off, we need to keep the frame buffer and the multiplier buffer.

The Complete Scheme

Figure 3 illustrates the complete scheme of applying this low computation cost Greyworld technique for on board video capture. In general, given an input frame $I(i)$, we first examine whether the frame is at the boundary of a luminance change by applying the adaptive luminance change detection scheme. If it is, we perform the multiplier computation to refresh the multiplier buffer and use such a multiplier to modify the entire frame for white balance. Otherwise, the input frame is considered to share the same luminance with the previous frame, and we use the multiplier in the buffer to perform block-based modification.

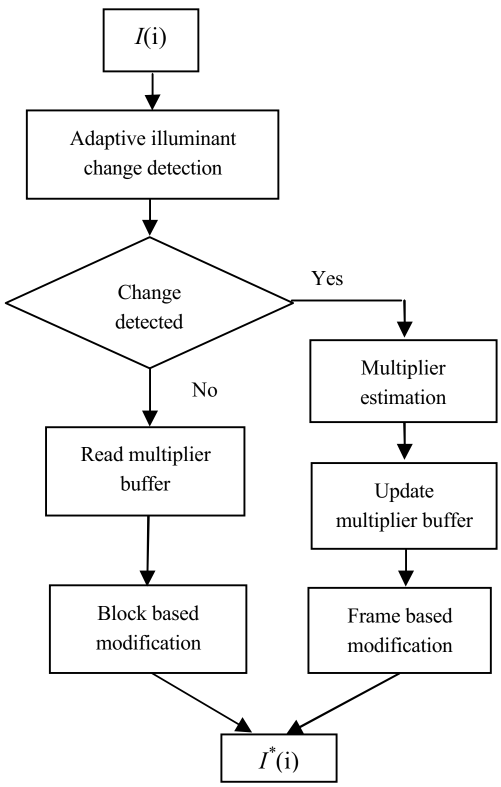

Figure 3 Proposed low computation cost Greyworld-based white balancing approach for on-board video capture

Experimental Results

(a) Original video frames (b) After white balancing using our proposed approach

Figure 4 Test sequence

Conclusions

This paper has addressed a low computation cost Greyworld-based white balance technique for on board video capture. As compared with other Greyworld-based white balance technique for video capture, the proposed technique has the distinguish advantage on the computation cost. By taking the temporal correlation of frames and consistency of illumination into account, the proposed technique can lower the amount of multiplication operations involved in the color modification.

References

[1] Yoon Kim, June-Sok Lee, A.W. Morales, and Sung-Jea Ko, "A Video Camera System with Enhanced Zoom Tracking and Auto White Balance," IEEE Transactions on Consumer Electronics, vol. 48, no. 3, pg. 428-434. (2002)

[2] June-Sok Lee, You-Young Jung, Byung-Soo Kim, and Sung-Jea Ko, "An Advanced Video Camera System with Robust AF, AE, and AWB Control," IEEE Transactions on Consumer Electronics, vol. 47, no. 3, pg. 694-699. (2001)

[3] HongJiang Zhang; Yihong Gong, S.W. Smoliar, Shuang Yeo Tan, "Automatic Parsing of News Video," Proceedings of the International Conference on Multimedia Computing and Systems, pg. 15-19. (1994)

[4] Arnon Amira, Gal Ashourb, and Savitha Srinivasan, "Automatic Generation of Conference Video Proceedings," Journal of Visual Communication and Image Representation, vol. 15, No. 3, pg. 467-488. (2004)

[5] Homer H. Chen, Chun-Hung Shen, and Pei-Shan Tsai, "Edge-Based Automatic White Balancing with Linear Illuminant Constraint," Proceedings of SPIE International Conference on Video Communications and Image Processing (VCIP), San Jose, CA, USA, vol. 6508, pg. 65081D-1 – 65081D-12. (2007)

Author Biography

Peng Wu received his PhD in Eletrical Engineering from University of California, Santa Barbara (2001). Since then he has worked in the Imaging Technology Department at HP Labs in Palo Alto, CA. His work has focused multimedia content analysis and multimedia systems.

Yuxin (Zoe) Liu received her PhD in electrical engineering from Purdue University, West Lafayette, IN (2004). She was with Nokia Research Center in Irving, TX from Aug. 2004 through Dec. 2005 as a Research Engineer, and was with Sun Labs in Menlo Park, CA from Dec. 2005 through Aug. 2006 as a Staff Engineer. Since Aug. 2006 she has been with HP Labs in Palo Alto, CA as a Research Scientist. Zoe's major research interests have been focused on video compression and media analysis.