Automatic Detection of Red-Eye Artifacts in Digital Color Photos

A. Mufit Ferman
Sharp Laboratories of America, Inc., Camas, WA 98607 USA
mferman@sharplabs.com

ABSTRACT

Red-eye is a flash-induced artifact that frequently occurs in photographs. This paper presents a novel method for unsupervised discovery of red-eye pixels in a digital image. Detection of face and/or skin regions in the image is not required by the algorithm. Analysis is performed primarily in the hue-saturation-value (HSV) color space. A flash mask, which defines the regions where red-eye regions may be present, is first extracted from the brightness component. Subsequent processing on the other color components prunes the number of candidate regions that may correspond to red-eye. Algorithm performance is demonstrated on content from multiple digital cameras with different resolutions.

Keywords: Image analysis, image segmentation, photography

1. Introduction

Red-eye is a common phenomenon that occurs during flash photography. In an environment where a flash is needed to illuminate the subject, the subject's pupils are dilated due to the low ambient illumination. Light from the flash can then enter the eye through the pupil and be reflected off the blood vessels at the back of the retina. This reflection is recorded by the camera if the geometry of the camera lens, the flash, and the subject's eye is just right, rendering the captured image unpleasant and objectionable to viewers.

A number of methods have been proposed in the literature for detecting and/or removing red-eye artifacts [1-4]. Most of these methods are either (i) supervised; i.e. they require the user to manually identify the subregions in an image where the artifacts are observed, or (ii) dependent on skin/face and/or eye detection to find the areas of interest. In this paper we propose an unsupervised technique that uses low-level image features to locate the red-eye pixels in a digital image, thereby eliminating the need to detect the face and/or skin regions in an image. Another advantage of the approach is that it limits processing to those areas in the image that are affected by the flash illumination, since the red-eye phenomenon is a direct result of flash use. The computational overhead is therefore reduced considerably. Basic processing techniques such as median filtering and morphological operations, as well as color- and shape-based constraints, are utilized repeatedly and in succession to reduce the number of candidate regions that may correspond to red-eye. Furthermore, each component of the image color space is analyzed separately, and the results are merged to yield a more reliable output.

2. Algorithm Overview

Figure 1 illustrates the flowchart of the proposed method. The input image is first converted into the HSV (hue/saturation/value) color space, where the rest of the processing occurs. The HSV space possesses desirable properties that are exploited by the algorithm to progressively refine the set of candidate red-eye regions. Each channel of the image is processed and analyzed in different ways to accurately identify the red-eye artifacts. The major steps of the algorithm are reviewed in the following sections.

2.1 Construction of the Flash Mask

As discussed earlier, red-eye artifacts occur as a direct consequence of flash use. The red-eye detection method therefore focuses only on those regions of the image that have been affected (i.e. illuminated) by a flash. In order to detect these areas of interest, the histogram of the value component (V) is analyzed. V is defined at each pixel location i as max(R_i,G_i,B_i), and hence can be used to measure the brightness of the scene. A flash-illuminated scene is expected to have a roughly bimodal luminance distribution, with a brightly-lit foreground and a darker background. A thresholding operation is therefore applied to the V component I_v of the image to separate the flash-illuminated foreground from the background. Pixels that exceed a threshold value T_f comprise the candidate foreground region. The value of T_f is selected using Otsu's method [5], a well-known thresholding technique.

978-1-4244-1764-3/08/$25.00 ©2008 IEEE

A 3x3 median filter is subsequently applied to the foreground pixels to eliminate isolated pixels. The remaining 'on' pixels are then grouped into contiguous regions using a connected component labeling algorithm, and regions with areas smaller than a predetermined threshold are discarded. The convex hull of each remaining region is subsequently computed and a binary mask that comprises the union of the convex hulls is constructed to yield the flash mask M_f. Various stages in the construction of M_f are depicted in Figure 2.

2.2 Analysis of Hue and Saturation Components

The flash mask M_f represents the areas in the input image that may contain red-eye artifacts; therefore, the rest of the processing is restricted to the regions identified by M_f. The hue component of the image, I_h, is masked with M_f to yield a masked version, I_h^m. Hue corresponds to the dominant color of a pixel, and it is represented as an angle on the unit circle between 0° and 360° [6]. The hue angles in face regions (where $R \geq G \geq B$) fall into the [0°, 60°] range, while for red-eye artifacts and surrounding areas (where $R \geq B \geq G$) the observed hue angles are in the [300, 360) interval. As a result, when the hue values are mapped to an appropriate interval for display (e.g., to [0,1] or [0°, 255°]), red-eye locations are observed to always appear as light, contiguous regions on darker backgrounds, as shown in Figure 3(a). This property is exploited by thresholding I_h^m to eliminate the dark areas and thus reduce the area that needs to be analyzed for red-eye artifacts. The value of the threshold T_h is set empirically to 0.125 in the current implementation.

After M_h is obtained, several post-processing operations are applied to refine it. These include median filtering and morphological operations including dilation and closing. The selected pixels in M_h are then grouped into contiguous regions using a connected component labeling algorithm, and several features are computed for each labeled region. Specifically, we consider the area, aspect ratio, and extent of every region to determine the likelihood that the region is a red-eye area. *Extent* is defined as the ratio of the total area of the region (i.e. the number of pixels in the region) to the number of pixels in the smallest bounding box for the region. Regions whose areas and/or aspect ratios fall outside predetermined ranges, or whose extent values are below a specified threshold, are discarded. The aspect ratio test allows us to eliminate regions that are elongated; the aspect ratio of a candidate red-eye region is expected to be in the interval (0.33, 2). Finally, if the extent of a region is less than 0.33, the region is removed from the list of candidate red-eye locations. Various stages in the construction of the hue mask M_h are shown in Figure 3.

Next, the information in the saturation component is utilized to further refine the list of candidate red-eye regions. Pixels in the red-eye regions often have high saturation values, as seen in Figure 4(a). Furthermore, the local variation in the saturation component is highly pronounced around the red-eye regions. To exploit this property, the standard deviation of the saturation component is computed for each pixel using a local neighborhood. Pixels that are likely to be red-eye artifacts are identified by a thresholding operation, which yields the saturation mask M_σ. The intersection of M_h and M_σ is then computed to yield a mask $M_{h\sigma}$ that represents the locations where the red-eye artifacts are most likely to occur. As in earlier stages of the algorithm, $M_{h\sigma}$ is refined by means of several post-processing steps, which include median filtering and morphological operations such as dilation and closing. The selected pixels in $M_{h\sigma}$ are grouped into contiguous regions using a connected component labeling algorithm, and several shape-based features are computed for each labeled region. Specifically, we compute the eccentricity and circularity of every region. *Eccentricity* is defined as the ratio of the distance between the foci of the ellipse that has the same second-moments as the region and its major axis length. The value of eccentricity varies between 0 and 1; the higher the eccentricity value, the closer to a line segment the region is. *Circularity*, as the name implies, is a measure of how closely a region resembles a circle, and is defined as the ratio of the square of the region perimeter to the area of the region. These properties are used to determine the likelihood that a particular region contains red-eye artifacts.

The final stage of the algorithm involves color-based analysis of the remaining regions to determine which pixels are strongly red. In the current implementation, we utilize the RGB values of the pixels in each candidate region to determine whether the region contains a red-eye artifact. The RGB values can be computed directly from the available HSV components by means of a simple transformation (or the original input image can be used directly, if it is available). For every region, we compute the mean of each primary. We then observe whether (i) the mean red value is less than a specified threshold, or (ii) the ratio of the means of the green and blue components is below another predetermined threshold. Any region that satisfies either of the above criteria is discarded, and the remaining regions are declared red-eye artifact locations. The individual pixels that require correction within these regions are identified through an analysis of the color properties of the individual pixels. This analysis includes thresholding based on pixel color values and clustering/region growing based on color similarity. The final output of the algorithm is a mask that identifies the individual pixels in the image that require red-eye correction, as shown in Figure 4(d).

3. Experimental Results

The proposed red-eye detection algorithm has initially been tested on 20 test images that have been compiled from six different digital cameras. Table 1 lists the model name and resolution for each of the cameras. Using content from multiple cameras allowed us to determine the robustness of the approach. To eliminate the effects of different image sizes on various algorithm properties, every input image was scaled down to the same dimensions (in this case, 960x1280). Each input image was studied by an expert, who manually labeled the red-eye regions in a given image. The images all contained red-eye artifacts of various size. The same parameter set was utilized for all input images.

Algorithm accuracy was measured using precision and recall, commonly used to for assessment of information retrieval systems [7]:

$$Precision = \frac{Correct}{Correct + False\ Alarms},\quad Recall = \frac{Correct}{Correct + Missed}$$

Precision and recall jointly rate the performance of a classification/retrieval technique. A consistent method produces precision and recall values that are close to unity.

Table 2 summarizes algorithm performance on the initial test set. Of the 52 total red-eye regions in the set, only 2 are missed by the algorithm. False alarm rate is relatively high, as reflected by the precision value, but acceptable. The main cause for the false alarms was observed to be specular highlights due to flash illumination. Additionally, in some instances sufficiently small regions with a strong red hue were falsely labeled as red-eye artifacts.

4. Conclusions and Directions for Further Research

In this paper we have presented an unsupervised method for detection of red-eye artifacts in digital color photos. Initial performance evaluation over content from multiple cameras has shown that the algorithm is robust and has high detection accuracy. We plan to improve the algorithms and reduce the number of false alarms by investigating how the flash mask and its properties can be used for determining whether the input image was captured with flash.

Camera Model	Image Resolution
FujiFilm FinePix 1300	960x1280
FujiFilm FinePix 4900 Zoom	1800x2400
Nikon Coolpix E950	1024x768
Nikon Coolpix E990	2048x1536
Konica-Minolta Dimage Xg	2048x1536
Canon PowerShot A710 IS	3072x2304

Table 1 Model names and resolutions of the six cameras that were used in testing.

Correct	Missed	False Alarms	Precision	Recall
50	2	18	0.7353	0.9615

Table 2 Algorithm performance on the initial test set, which contained a total of 20 photos and 52 red-eye regions.

References:

[1] Hardeberg, J.Y., "Digital Red Eye Removal," in J. of Imaging Sci. and Technology, vol. 46, no. 4, pp. 375-381, July/August 2002.

[2] Held, A., "Model-Based Correction of Red-Eye Defects," in *Proc. 10th Color Imaging Conference (CIC)*, pp. 223-228, Scottsdale, Arizona, Nov. 2002.

[3] Schildkraut, J. S. and Gray, R., "A Fully Automatic Redeye Detection and Correction Algorithm," in *Proc. ICIP2002*, vol. I, pp. 801-803, Rochester, NY, Sep. 2002.

[4] Gaubatz, M. and Ulichney, R., "Automatic Red-Eye Detection and Correction," in *Proc. ICIP2002*, vol. I, pp. 804-807, Rochester, NY, September 2002.

[5] Otsu, N., "A thresholding selection method from gray-level histogram," in *IEEE Trans. Syst. Man Cybernet.* 9(1), 62-66, 1979.

[6] Foley, J. D. et al., *Computer Graphics: Principles and Practice (2nd ed.)*, pp. 592-593, Addison-Wesley Professional, 1996.

[7] Salton, G., *Automatic Information Organization and Retrieval*, McGraw-Hill, NY, 1968.

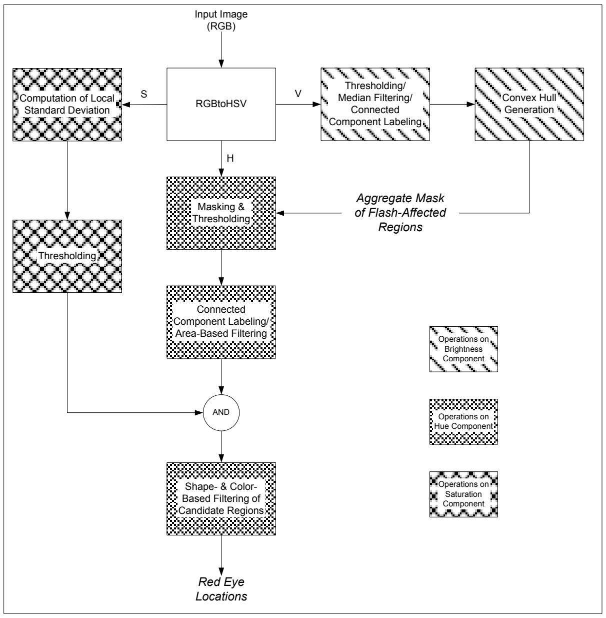
Input Image (RGB)
S
V
H
RGBtoHSV
Computation of Local Standard Deviation
Thresholding/ Median Filtering/ Connected Component Labeling
Convex Hull Generation
Masking & Thresholding
Aggregate Mask of Flash-Affected Regions
Thresholding
Connected Component Labeling/ Area-Based Filtering
AND
Shape- & Color-Based Filtering of Candidate Regions
Red Eye Locations
Operations on Brightness Component
Operations on Hue Component
Operations on Saturation Component

Figure 1 Flowchart of the automatic red-eye detection system.

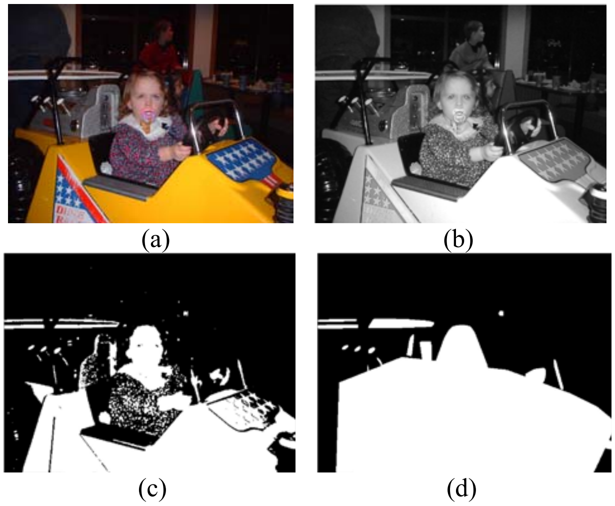
(a) (b) (c) (d)

Figure 2 Construction of the flash mask for the input image: (a) Original input image; (b) value component I_v; (c) flash mask I_f obtained after thresholding by T_h; (d) final mask image, M_f.

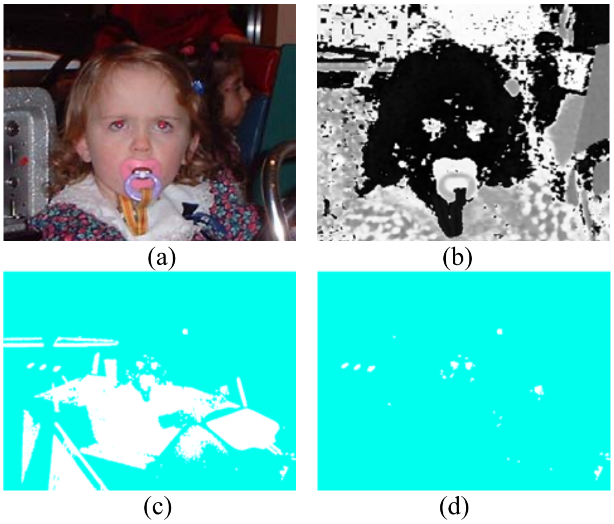
(a) (b) (c) (d)

Figure 3 Identification of candidate red-eye regions using M_f and the hue component: (a) Close-up of the red-eye regions of the image; (b) hue representation of the same region; (c) hue component I_h^m after thresholding and post-processing; and (d) final candidate red-eye locations after connected component labeling and area- and shape-based filtering. (Unprocessed regions are shown in blue.)

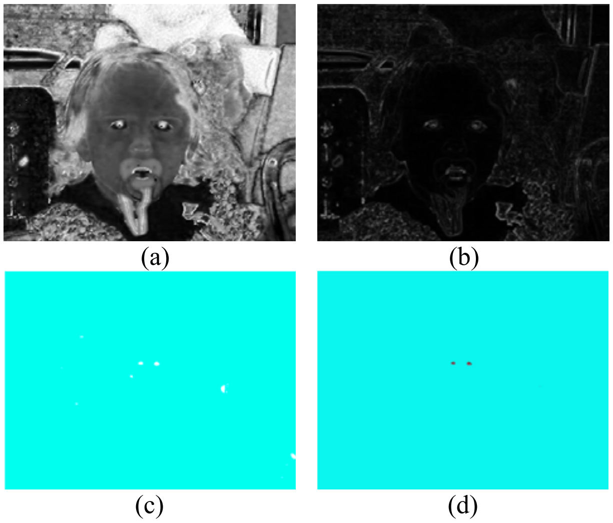
(a) (b) (c) (d)

Figure 4 Identification of the red-eye artifact regions using the saturation component: (a) Saturation component I_s; (b) standard deviation map $I_{s\sigma}$; (c) candidate red-eye regions obtained by thresholding $I_{s\sigma}$ and intersecting M_h and M_σ, and (f) final red-eye locations identified after shape-based filtering and color analysis. Unprocessed background regions are shown in blue.