

AUTOMATIC RED-EYE DETECTION AND CORRECTION

Matthew Gaubatz

Cornell University
gaubatz@ece.cornell.edu

Robert Ulichney

Hewlett-Packard Co.
Bob.Ulichney@hp.com

ABSTRACT

“Red-eye” is a phenomenon that causes the eyes of flash photography subjects to appear unnaturally reddish in color. Though commercial solutions exist for red-eye correction, all of them require some measure of user intervention. A method is presented to automatically detect and correct red-eye in digital images. First, faces are detected with a cascade of multi-scale classifiers. The red-eye pixels are then located with several refining masks computed over the facial region. The masks are created by thresholding per-pixel metrics, designed to detect red-eye artifacts. Once the red-eye pixels have been found, the redness is attenuated with a tapered color desaturation. A detector implemented with this system corrected 95% of the red-eye artifacts in 200 tested images.

1. INTRODUCTION

In flash photography, red-eye is the appearance of an unnatural red hue around a person’s pupils. When a flash is needed to illuminate a subject, the ambient illumination is generally low, and the subject’s pupils will be dilated. Light from the flash can thus reflect off the blood vessels in the subject’s retina and appear red in color. If the angle formed by the flash, the back of the subject’s eye, and the camera lens is sufficiently small, this reddish light will be recorded by the camera.

A popular solution for cameras with a small lens-to-flash distance is the use of one or more pre-exposure flashes. A pre-exposure flash will contract the subject’s pupil diameter, thus reducing the chance that light reflected off the retina will reach the lens. The drawback of this approach is the cost in power. A flash consumes more power than any other aspect of image capture; additional flashes further reduce the battery life. Furthermore, this flash will sometimes reduce, but not eliminate, the red-eye artifacts. As a result, a software solution is desirable.

To date, there have been few results published specific to automatic red-eye detection. However, extensive research has been done on the detection of facial features. A natural approach to eye detection is to divide the task into two parts, 1) the detection of faces within an image and 2)

the detection of eyes within a face. Such techniques have been presented by Zhang and Lenders, as well as Xin, *et al.* [1, 2]. Promising results in each of these individual areas, face and eye detection, have also been achieved. Neural network, support vector machine, wavelet and eigenvector techniques have been used to build face detectors [3, 4, 5]. Rowley *et al.* and Viola and Jones have made significant improvements in detection speed by cascading a number of different detectors [3, 6]. Kawaguchi, Rizon, and Hikada have presented techniques to locate eyes using intensity information [7, 8].

Because color is critical to the removal of red-eye artifacts, chroma as well as intensity information can be used to put constraints on the search space. A detection procedure is presented that uses a face detector to find faces, then uses color, intensity and size information to detect red-eye in images of varying quality. The contributions of this paper include a flexible, efficient method for detecting red-eye artifacts in facial images, a minimally invasive corrective procedure designed to correct red-eye in a visually pleasing manner, and a fully automatic red-eye correction algorithm requiring no user intervention.

2. ALGORITHM OVERVIEW

The algorithm consists of three stages: 1) face detection, 2) red-eye detection, and 3) red-eye correction. Initially, images are analyzed by the Viola-Jones face detection system, described in [6]. Detected facial regions are then processed by the red-eye detector. It is assumed that input images are in YUV or RGB space, and that conversions required by the algorithm are available.

The task of effectively detecting red-eye is similar to the task of effectively describing red-eye. In certain cases, redness alone can be used to describe red-eye artifacts. Unfortunately, this property does not robustly describe red-eye artifacts in all images. Some eye detection schemes use the Hough transform to measure how well candidate features match the expected shape of an eye [2, 7, 8]. However, pictures are sometimes out of focus, and the subject’s features appear blob-like, without much definitive shape. Along with intensity and redness, eye size (which varies lit-

tle with image quality) is used to define per-pixel metrics to emphasize red-eye regions. The red-eye detector evaluates a series of these metrics to eliminate pixels that do not belong to regions of red-eye. A list of candidate red-eye locations is passed to the red-eye correction system.

Once the red-eye pixels have been located, the correction processing involves a simple desaturation of color. A corrective mask is constructed to rescale the chrominance values of red-eye region pixels to reflect a neutral color. Without prior knowledge about the subject's eye color, removing the unnatural redness is the least invasive type of correction.

3. FACE DETECTION

The Viola-Jones face detection system uses a multi-scale multi-stage classifier, which operates on image intensity information. A cascade of thirty-eight simple multi-scale feature detectors is used to eliminate image regions that do not contain faces. The face detector adds several useful features to the overall red-eye correction procedure. First, the face detector is very fast. Secondly, the face detector narrows the search for red-eye artifacts to the facial regions of subjects within an image by generating a list of bounding boxes that contain frontal upright faces. This fact can be used to deduce information about the location and size of possible red-eye artifacts. Because the detector finds upright faces, the search can be narrowed to the upper half of the face, as in Fig. 1. Also, the dimensions of a bounding box are usually within 25% of the dimensions of the corresponding face. Because the ratios of eye width and height to face width and height are similar in most faces, the face detector gives an indication of expected eye size in a detected face. This size information is very useful for locating red-eye artifacts. In addition, the summed-area-table image representation, which is used by the face detector, can also be used to increase the speed of expensive filtering operations used red-eye detection algorithm [9].

4. RED-EYE DETECTION

The structure of the red-eye detector is similar to the structure of the face-detector. The red-eye detector uses a series of four feature detectors (metrics) to mask non-red-eye pixels. These metrics emphasize redness and several types of changes in luminance and redness. The color variation metric is computed first. The resulting values are thresholded to create a mask that eliminates most non-red-eye pixels. If necessary, other metrics are used to create masks that refine the estimates of candidate red-eye pixels detected by the color variation mask.

Various metrics are employed because red-eye occurs in different degrees of severity in pictures varying in qual-



Fig. 1. Upper half of detected face.

ity. The system is flexible in that a new metric can easily be added to handle a new general case. For instance, the case in which the entire eye is washed out by reflected light is not handled by this system, primarily because the corrected eye often looks as unnatural as the red-eye artifact. However, another metric designed to emphasize larger, brighter patches of redness could be included in the detection algorithm, described below.

The color variation mask is computed over the facial region (see Fig. 2). A test is performed to verify whether or not the mask reveals two eye-sized clumps of pixels. If so, detection is complete. Otherwise, the test is repeated on the pixels revealed by the combination of the color variation mask and a new mask (derived from a new metric). This step repeats until either the test is passed, or no additional metrics are available. If no metrics remain, the correction procedure is applied around the pixels retained by the combination of all computed masks. The metrics are presented in the order they are computed during detection.

4.1. Color variation

In a facial image, eye regions are usually characterized by sharp changes in luminance and chrominance within a small area of a particular size. To measure such a quantity at a location (x, y) , the following quantity is computed:

$$\text{variation}I(x, y) = \frac{1}{|R_{x,y}|} \sum_{\mathbf{r} \in R_{x,y}} \left\| \mathbf{r} - \frac{1}{|P_{\mathbf{r}}|} \sum_{\mathbf{p} \in P_{\mathbf{r}}} \mathbf{p} \right\|_2^2,$$

where \mathbf{p} , \mathbf{r} are pixels (points in YUV space), $R_{x,y}$ is an eye-shaped region centered on (x, y) , $P_{\mathbf{r}}$ is a pupil-sized region centered around pixel \mathbf{r} , and $|\cdot|$ denotes the number of elements in a set. Because the face detector reveals the size of eyes in the facial region, the sizes of $P_{\mathbf{r}}$ and $R_{x,y}$ can be estimated. Roughly speaking, this metric emphasizes eye-shaped regions that include pupil-sized variations in color.

4.2. Redness

There is no best measure of redness for red-eye pixels, though empirically the red-eye artifact pixels usually contain purer

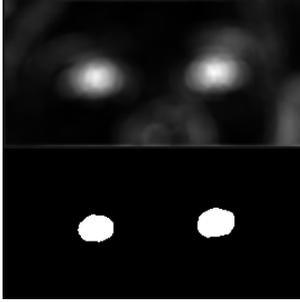


Fig. 2. Example metric and resulting mask.

hues of red compared to the to the rest of the pixels in a given image. More precisely, the ratio of energy of the red component to that of the remaining pixel energy in a red-eye pixel is generally higher than the same ratio in other pixels. For a pixel at location (x, y) , the following ratio is a measure of the pixel's redness:

$$redness(x, y) = \frac{R(x, y)^2}{G(x, y)^2 + B(x, y)^2 + K},$$

where constant K is used to avoid singularities.

4.3. Redness variation

If a red-eye artifact is visible to a human observer, it is likely that the artifact is surrounded by pixels of a different degree of redness, creating an oscillation in the redness measure. The height of an eye (and thus the size of the vertical oscillation in redness) usually falls between 5% and 8% of the height of the facial bounding box. Since the size of the face is known with respect to the sampling rate of the data, it is possible to estimate bounds on the frequency of this oscillation:

$$\begin{aligned} \text{minimum frequency} &\approx 1/(\text{maximum eye height}) \\ &\approx 1/(0.08 \times (\text{box height})) \\ \text{maximum frequency} &\approx 1/(\text{minimum eye height}) \\ &\approx 1/(0.05 \times (\text{box height})) \end{aligned}$$

A filter can be designed to pass this frequency (e.g. with the Parks-McClellan algorithm [10]). The redness metric plane is filtered one-dimensionally in the vertical direction. The energy of this filtered signal is another measure of variation in redness:

$$\begin{aligned} variationII(x, y) &= |redness(x, y) * g(x, y)|^2 \\ &= |redness(x, y) * g(y)|^2 \end{aligned}$$

4.4. Glint

Another feature that aids in the detection of red-eye is the glint caused by a camera flash. Since a flash is the cause

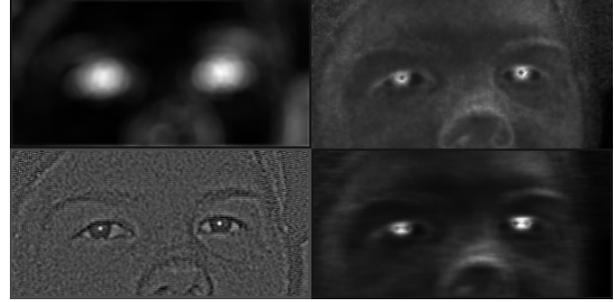


Fig. 3. Color variation, redness, redness variation and glint metrics (clockwise from top left).

of red-eye, a reflection of the flash is often highly visible in the eyes of the subject. The Laplacian operator can easily locate such sharp changes in brightness. However, it will also emphasize pixels with excessive noise, which is undesirable. Hence, a low-pass-filtered version of the Laplacian, computed over the luminance plane, Y , can be used to detect eye glint at a point (x, y) in the image:

$$glint(x, y) = (-\nabla^2 Y(x, y)) * h(x, y),$$

where $h(x, y)$ is a low-pass filter. These metrics, computed over the image in Fig. 1, are shown in Fig. 3.

5. RED-EYE CORRECTION

Once a candidate red-eye location is detected, the artifact can be corrected by removing the color of all “excessively red” pixels within an eye-sized neighborhood. A pixel at location (x_0, y_0) is considered excessively red if $redness(x_0, y_0)$ exceeds a certain threshold. Handling the artifact in this manner will remove the redness, but can introduce other undesirable artifacts. If pixels are desaturated based only on their corresponding redness values, reddish pixels outside the eye could be desaturated. Also, a patch of eye pixels devoid of color can look even more unnatural than a patch of slightly reddish pixels, especially if the picture has a high mean luminance with a small dynamic range of color values. Finally, a visible hard boundary between corrected and uncorrected pixels can have a displeasing look.

Several steps are taken that improve the visual quality of the corrected image. Eye boundaries are marked by changes in luminance. Thus, pixels associated with a significant change in luminance are removed from the edge of the corrective mask, to prevent the desaturation of non-red-eye pixels. Undesired effects resulting from hard color desaturation are mitigated by desaturating pixels by an amount proportional to their redness. Perceived edges due to hard decision boundaries between corrected and uncorrected pixels are reduced by tapering the boundaries. This process, illustrated by Fig. 4, yields a patch centered on the red-eye

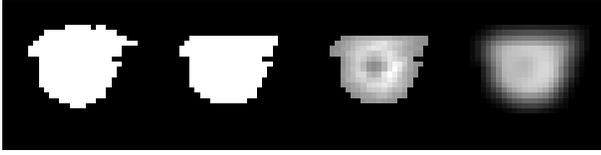


Fig. 4. Correction factor refinement stages.

artifact, with a “correction factor” computed for each pixel. The correction factor, $p(x, y) \in [0, 1]$, is the percentage by which the color of pixel in the patch is desaturated:

$$\begin{aligned} Y(x, y)_{corrected} &= Y(x, y)_{orig} \\ Cr(x, y)_{corrected} &= (1 - p(x, y)) * Cr(x, y)_{orig} \\ Cb(x, y)_{corrected} &= (1 - p(x, y)) * Cb(x, y)_{orig} \end{aligned}$$

This correction process results in a less invasive correction. A comparison of the corrected and uncorrected image is illustrated by Fig. 5.

6. RESULTS AND CONCLUSIONS

This system was tested on a collection of 200 photographs with faces, about half of which included subjects with red-eye. Images in the test bed consisted of scanned photographs, digital photographs, and images pulled off the web. The pictures varied in quality and size, with subjects at a variety of distances, focused and unfocused. Red-eye artifacts in the test bed ranged from slightly red pupils to bright red orbs. The algorithm successfully removed about 95% of the offending red-eye artifacts from images in the test bed. On a computer with a 1 GHz processor, correcting a 1 megabyte image took several seconds. The face detector only returned one false positive non-face, and red-eye detector found no red-eye this non-face region. In addition, the red-eye detector produced only one false positive red-eye detection.

Most of the artifacts missed by the system occurred in very small faces, which were often in the background. Because the detection algorithm is flexible, performance can be improved with the addition of a metric tailored for smaller images. In many cases, red-eye was detected by the first metric evaluated, color variation. Intuitively, red-eye artifacts are defined in terms of color. However, this color varies in different images. Properties of size and change in color provided useful alternate detection criteria, and greatly reduced the number of false red-eye detections.

7. REFERENCES

[1] L. Zhang and P. Lenders, “Knowledge-based eye detection for human face recognition,” in *Fourth International Conference on Knowledge-based Intelligent*

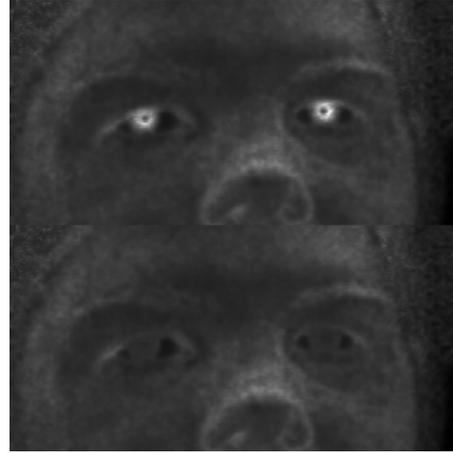


Fig. 5. Redness metric (before and after correction).

Systems and Allied Technologies, 2000, vol. 1, pp. 117–120.

- [2] L. Xin, X. Yanjun, and D. Limin, “Locating facial features with color information,” in *ICSP '98 Proceedings*, 1998, vol. 2, pp. 889–892.
- [3] H. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 10, pp. 22–38, 1998.
- [4] H. Schneiderman and T. Kanade, “A statistical method for 3d object detection applied to faces and cars,” in *International Conference on Computer Vision*, 2000.
- [5] D. Roth, M. Yang, and N. Ahuja, “A snowbased face-detector,” in *Neural Information Processing 12*, 2000.
- [6] P. Viola and M. Jones, “Robust real-time object detection,” Tech. Rep. 1, Compaq Cambridge Research Laboratory, February 2001.
- [7] T. Kawaguchi, D. Hikada, and M. Rizon, “Detection of the eyes from human faces by hough transform and separability filter,” in *ICIP 2000 Proceedings*, 2000, pp. 49–52.
- [8] M. Rizon and T. Kawaguchi, “Automatic eye detection using intensity and edge information,” in *Proceedings of TENCON 2000*, September 2000, pp. 24–27.
- [9] F. Crow, “Summed-area tables for texture mapping,” in *SIGGRAPH*, 1984, vol. 18(3), pp. 207–212.
- [10] J. H. McClellan, T. W. Parks, and L. R. Rabiner, “A computer program for designing optimum fir linear phase digital filters,” *IEEE Trans. Audio Electronics*, vol. AU-21, pp. 506–526, December 1973.