

Detection and Correction of Red Eye in Digital Photograph

Swati S. Deshmukh M.E. [IT], Sipna COET, Amravati, Maharashtra
Dr. A.D. Gawande ,Sipna COET, Amravati, Maharashtra
Prof. A.B.Deshmukh ,Sipna COET, Amravati, Maharashtra

Abstract—Red-eye effect often appears in photographs while taking pictures with flash. Flash light passing through pupil is reflected on the blood vessels, and arrives at a camera lens. This causes red-eyes in photographs. For Removing red-eyes in digital photographs many algorithms have been proposed. It can be removed by software available in market but most of them are manual .Here is proposed work which is automatic one. This work proposes a red-eye removal algorithm using in-painting which is largely composed of three parts: face detection, red-eye detection and red-eye correction. Face regions are detected first for red-eye detection. After that, red-eye regions are removed using in-painting. Next, pupils are painted to a proper circular shape. The proposed algorithm can be tested with large number of photographs with red-eye effect and the proposed algorithm effectiveness can be compared with the conventional algorithms.

Index Terms—Red eye effect, red eye correction, red eye detection.

I. INTRODUCTION

What is red eye effect?

Red eye effect occurs because the light of the flash occurs too fast for the pupil to close much of the very bright light from the flash passes into the eye through the pupil, reflect off the fundus at the back of the eyeball and out through the pupil. The camera records this reflected light. The main cause of the red color is the ample amount of blood in the choroid which nourishes the back of the eye and is located behind the retina. The blood in the retinal circulation is far less than in the choroid, and plays virtually no role. The eye contains several photostable pigments that all absorb in the short wavelength region, and hence contribute somewhat to the red eye effect. The lens cuts off deep blue and violet light, below 430 nm (depending on age), and macular pigment absorbs between 400 and 500 nm, but this pigment is located exclusively in the tiny fovea. Melanin, located in the retinal pigment epithelium (RPE) and the choroid, shows a gradually increasing absorption towards the short wavelengths. But blood is the main determinant of the red color, because it is completely transparent at long wavelengths and abruptly starts absorbing at 600 nm. The amount of red light emerging from the pupil depends on the amount of melanin

in the layers behind the retina. This amount varies strongly between individuals. Light skinned people with blue eyes have relatively low melanin in the fundus and thus show a much stronger red-eye effect than dark skinned people with brown eyes. Following is the Fig 1. Shows Light reflecting from eye.

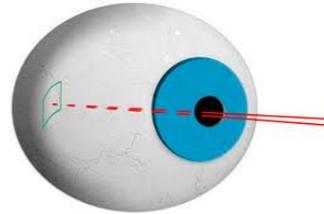


Fig. 1 Light reflecting from eye.

How to prevent Red eye effect?

When such light reflected from eye it causes red pupil to appear. Such red pupil make image useless. The red-eye effect can be prevented in a number of ways. We can use bounce flash in which the flash head is aimed at a nearby pale colored surface such as a ceiling or wall or at a specialist photographic reflector. This both changes the direction of the flash and ensures that only diffused flash light enters the eye. The flash can be placed away from the camera's optical axis ensures that the light from the flash hits the eye at an oblique angle. The light enters the eye in a direction away from the optical axis of the camera and is refocused by the eye lens back along the same axis. Because of this the retina will not be visible to the camera and the eyes will appear natural. Pictures can be taken without flash by increasing the ambient lighting, opening the lens aperture. Using the red-eye reduction capabilities built into many modern cameras. These precede the main flash with a series of short, low-power flashes, or a continuous piercing bright light triggering the pupil to contract.

Another way is to having the subject look away from the camera lens. Photograph subjects wearing contact lenses with UV filtering. Increase the lighting in the room so that the subject's pupils are more constricted. Professional photographers prefer to use ambient light or indirect flash, as the red-eye reduction system does not always prevent red eyes — for example, if people look away during the pre-flash. In addition, people do not look natural with small pupils, and direct lighting from close to the camera lens is considered to produce unflattering photographs.

Here we are using three main steps to remove red eye. First step is to detect face in image. Face detection is foremost the difficult task. Here face detection is done by using Viola Jones Face detection effective algorithm. Once the face is detected eye is detected in detected face and then eye is removed by using in painting method.

II. PREVIOUS WORKS DONE

A method was presented by M. Gaubatz and R. Ulichney [1] to automatically detect and correct redeye in digital images in their work "Automatic red-eye detection and correction." They told that 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 redeye pixels have been found, the redness is attenuated with a tapered color desaturation. But the face detector in their work 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 are often in the background. Because the detection algorithm is flexible, performance can be improved with the addition of a metric tailored for smaller images.

R. Schettini, F. Gasparini, and F. Chazli, [2] in "A modular procedure for automatic redeye correction in digital photos," used an adaptive color cast algorithm to correct the color photo. The phase used not only facilitates the subsequent of processing but, also improves the overall appearance of output image. A multi-resolution neural network approach has been used for mapping of candidate faces. They have reduced search space by using information about skin and face distribution. The overall performance of this method would be improved by increasing the efficiency of face detector and by introducing most geometric constraints.

Huitao Luo, Jonathan Yen and Dan Tretter[4] in work 'An Efficient Automatic Redeye Detection and Correction Algorithm' said that Adaboost has used to simultaneously select features and train the classifier. A new feature set is designed to address the orientation-dependency problem associated with the Haar-like features commonly used for object detection design. For each detected redeye, a correction algorithm is applied to do adaptive desaturation and darkening over the redeye region. In their work, the verification classifiers are trained in two stages: a single eye verification stage and a pairing verification stage. Adaboost [1] has used to train the classifiers because of its ability to select relevant features from a large number of object features. This is partially motivated by the face detection work of Viola and Jones [5]. However, in comparison to their work, their contributions come in three aspects. First, in addition to grayscale features, their detection algorithm utilizes color information by exploring effective color space projection and space conversion in designing object features. Second, he design a set of non-orientation-sensitive

features to address the orientation-sensitivity problem associated with the Haarlike features used in Viola and Jones' work. Third, their algorithm uses not only Haar-like rectangle features, but also features of different semantics such as object aspect ratio, percentage of skin tone pixels, etc.

The proposed redeye removal system by [4] contains two steps:

The red eye detection and the red eye correction. The detection step contains three modules: initial candidate detection, single eye verification and pairing verification. Among them, the initial candidate detection is a fast processing module designed to find all the red oval regions that are possibly red eyes. The single eye verification module verifies the redeye candidates using various object features, and eliminates many candidate regions corresponding to false alarms. Pairing verification further verifies the remaining redeye candidates by grouping them into pairs.

Jutta Willamowski, Gabriela Csurka [5] in work 'Probabilistic Automatic Red Eye Detection and Correction' uses contrast approach, their work does not require face detection, and thus, enables the correction of red eyes located on faces that are difficult to detect. A possible drawback is that it detects red eye candidates that do not correspond to real eyes. However, their method was able to reject or to assign a low probability to most of these locations which avoids introducing damaging artifacts. Some existing approaches use learning based red eye detection methods. Learning based methods rely on the availability of a representative training set.

In S. Ioffe, *Red eye detection with machine learning*, IICIP and L. Zhang, Y. Sun, M. Li, and H. Zhang, *Automated Red-Eye Detection and Correction in Digital Photographs*, IICIP 2004 this training set has to be manually collected, and the eyes properly cropped and aligned. Similarly, during detection or verification, test candidate patches or the test images have to be tested at varying size and orientation. In their approach, similarly to they constitute the training set automatically through the initial candidate detection step. It only requires labeling the detected candidates as *red eyes* or *non red eyes*. This has the advantage of "concentrating" the learning step on the differences between true and false positive patches. In contrast to H. Luo, J. Yen and D. Tretter, *An Efficient Automatic Redeye Detection and Correction Algorithm*, ICPR 2004, they introduced an additional distinction between false positives on *faces* and on *background*. The major advantage of their approach resides in combining *probabilistic* red eye detection and *soft* red eye correction. Most previous approaches adopt in the end a *hard* yes/no decision and apply either no or the maximal possible correction. In difficult cases, *hard* approaches make significant mistakes completely missing certain red eyes or introducing disturbing artifacts on non red eye regions. Their correction is often unnatural, e.g. resulting in a remaining reddish ring around the central corrected part of an eye. To obtain a more natural correction [5] introduces softness blurring the detected red eye region.

F. Volken, J. Terrier, and P. Vandewalle [6] in "Automatic red-eye removal based on sclera and skin tone detection," used the basic knowledge that an eye is

characterized by its shape and the white color of the sclera. Combining this intuitive approach with the detection of “skin” around the eye, they obtain a higher success rate than most of the tools. Moreover, their algorithm works for any type of skin tone. Further work is oriented towards improving the overall quality of the correction. It would be interesting to address the problems encountered for people with glasses, and to study more natural correction methods.

R.Ulichney and M.Gaubatz, in [8] “Perceptual-based correction of photo red-eye,” Presented a brief overview of facial image processing techniques it is given that the original pupil color of a subject is often unrecoverable a simple chrominance desaturation effectively removes the red hue from the artifact pixels. These are in turn combined with a fully automated procedure designed to minimize intrusive effects associated with pixel re-coloration.

III. PROPOSED WORK

Face detection

A face detection algorithm is used to limit the candidate region for red-eye detection. In this work, Viola and Jones’ algorithm is used to detect faces [9].

The features employed by the detection framework universally involve the sums of image pixels within rectangular areas. As such, they bear some resemblance to Haar basis functions, which have been used previously in the realm of image-based object detection.^[3] However, since the features used by Viola and Jones all rely on more than one rectangular area, they are generally more complex.. The value of any given feature is always simply the sum of the pixels within clear rectangles subtracted from the sum of the pixels within shaded rectangles. As is to be expected, rectangular features of this sort are rather primitive when compared to alternatives such as steerable filters. Although they are sensitive to vertical and horizontal features, their feedback is considerably coarser. However, with the use of an image representation called the integral image, rectangular features can be evaluated in *constant* time, which gives them a considerable speed advantage over their more sophisticated relatives. Because each rectangular area in a feature is always adjacent to at least one other rectangle, it follows that any two-rectangle feature can be computed in six array references, any three-rectangle feature in eight, and any four-rectangle feature in just nine.

Learning algorithm

The speed with which features may be evaluated does not adequately compensate for their number, however. For example, in a standard 24x24 pixel sub-window, there are a total of 45,396 possible features, and it would be prohibitively expensive to evaluate them all. Thus, the object detection framework employs a variant of the learning algorithm AdaBoost to both select the best features and to train classifiers that use them.

Cascade architecture

The evaluation of the strong classifiers generated by the learning process can be done quickly, but it isn’t fast enough to run in real-time. For this reason, the strong classifiers are arranged in a cascade in order of complexity, where each successive classifier is trained only on those selected samples which pass through the preceding classifiers. If at any stage in the cascade a classifier rejects the sub-window under inspection, no further processing is performed and continue on searching the next sub-window . The cascade therefore has the form of a degenerate tree. In the case of faces, the first classifier in the cascade – called the attentional operator – uses only two features to achieve a false negative rate of approximately 0% and a false positive rate of 40%. The effect of this single classifier is to reduce by roughly half the number of times the entire cascade is evaluated.

Eyes are located in specific area of the face region by the attribute of the face detection algorithm that is to be used. When the face region is divided into 20 cells, in four rows and five columns, it is observe that two eyes are mostly placed in the second-row. As shown in following Fig 2 face can be detected.

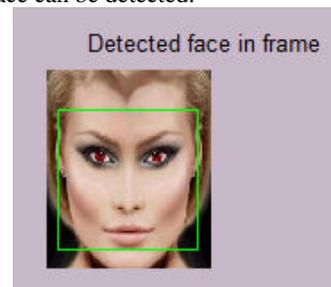


Fig. 2(a) Face Detection example

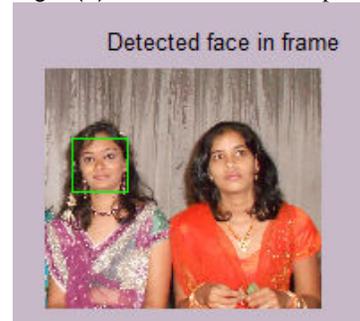


Fig. 2(b) Face Detection example

Most of faces are detected unless the face is occluded by other objects. And if face regions are found correctly, eyes are detected in the specific area. Red-eyes are detected by using features of red-eye extracted from the eye region.

IV. REDEYE REMOVAL USING INPAINTING

Eyes Extraction

According to observation eyes are located at particular area of face for every human being I used the face detected in face detection method. Once obtained the face as you can observe in above fig.2 Full face feature is taken by plotting rectangle over face and this rectangular face can be used to extract

eyes from rectangle by dividing it into half portion. Now upper half portion contain eyes located so only lower 20% area of upper half rectangle is used here to extract eyes as shown in fig.3



Fig. 3(a) Extracted eyes from face



Fig. 3(b) Extracted eyes from face



Fig.3(c) Extracted eyes from face

Image in-painting, also known as “image completion” or “disocclusion,” is the recovery of missing or corrupted parts of an image in a given region so that the reconstructed image looks natural. It has become a standard tool in digital photography for image retouching .

Although it is impossible to do justice to all of the many research efforts under way, most in-painting methods in the literature can be divided into two groups: geometry- and texture-oriented methods. Techniques of the two types were introduced at almost the same time; texture-oriented methods have proved better adapted to the production of natural-looking images, although ideas from both have contributed to the production of state-of-the-art results. Following fig. 4 shows red eye removed image using in-painting.

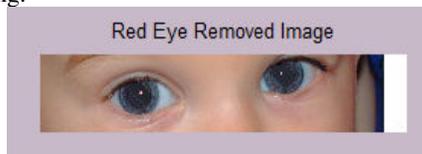


Fig.4(a) Red Eye Removal Using In-painting



Fig. 4(b) Red Eye Removal Using In-painting

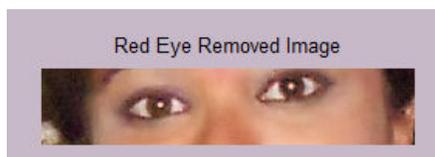


Fig. 4(c) Red Eye Removal Using Inpainting

V. RED EYE CORRECTION

Once red eyes are removed using in-painting it is most important to correct the in-painted pupil. Because due to flash whatever red eye effect is occurring will not necessarily give us circular red region. As per observation pupils are in circular shape so it is important to make circular pupil to look natural eye, for this purpose red eye correction is done here using a weighted average filter as shown in fig. 5

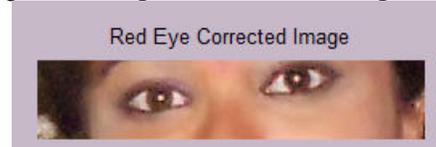


Fig. 5(a) Corrected image

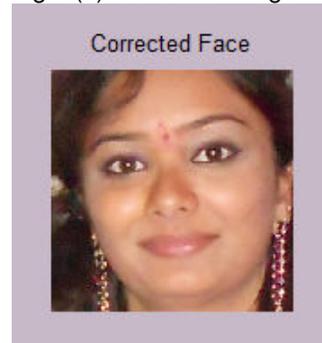


Fig. 5(b) Corrected image

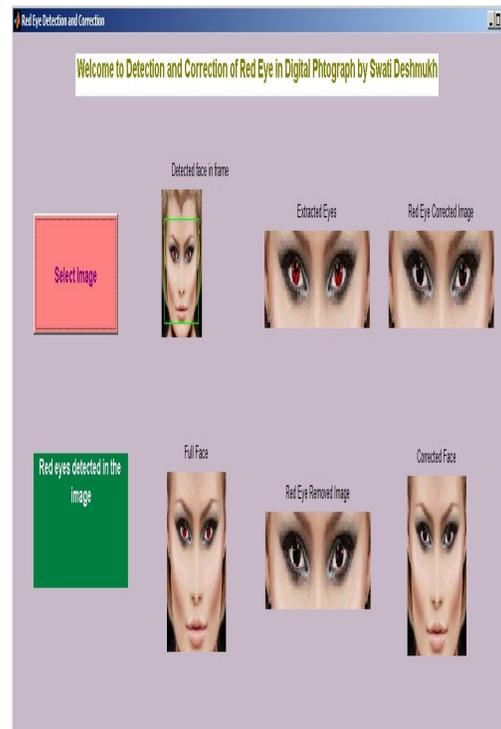


Fig. 5(c) Corrected image

VI. EVALUATION

In case of red-eye detection, performance is evaluated by detection rate and the number of false alarms (FAs). Detection rate is the ratio of the number of correctly found

red-eyes to the total number of red-eyes. And if the algorithm detects non-red-eye as red-eye, the count of FAs increases. In case of redevye correction, it is difficult to evaluate the performance quantitatively. Thus, we compare the result images. We compare the performance of the proposed algorithm with those of commercial softwares [17–19]. In comparison of detection performance, method of [17] is excluded since it does not offer automatic red-eye detection.

Table 1
COMPARISON OF RED-EYE
DETECTION.RESULTS

Algorithm	Detection rate (%)	False alarms
Method of [18]	62.8	23
Method of [19]	91.2	20
Method of [20]	92.9	15
Proposed work	93.75	8

REFERENCES

- [1] M. Gaubatz and R. Ulichney, "Automatic red-eye detection and correction," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 804–807, Rochester, NY, Sep. 2002.
- [2] R. Schettini, F. Gasparini, and F. Chazli, "A modular procedure for automatic redevye correction in digital photos," in *Proc. SPIE Conf. Color Imaging: Processing, Hardcopy, and Application*, vol. 5293, pp. 139–147, San Jose, CA, Jan. 2004.
- [3] S. Ioffe, "Red eye detection with machine learning," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 871–874, Barcelona, Spain, Sep. 2003.
- [4] H. Luo, J. Yen, and D. Tretter, "An efficient automatic redevye detection and correction algorithm," in *Proc. IEEE Int. Conf. Pattern Recognition*, vol. 2, pp. 883–886, Cambridge, UK, Aug. 2004.
- [5] J. Willamowski and G. Csurka, "Probabilistic automatic red eye detection and correction," in *Proc. IEEE Int. Conf. Pattern Recognition*, vol. 3, pp. 762–765, Hong Kong, China, Aug. 2006.
- [6] F. Volken, J. Terrier, and P. Vandewalle, "Automatic red-eye removal based on sclera and skin tone detection," in *Proc. European Conf. Color in Graphics, Imaging and Vision*, pp. 359–364, Leeds, UK, June 2006.
- [7] L. Zhang, Y. Sun, M. Li, and H. Zhang, "Automated red-eye detection and correction in digital photographs," in *Proc. IEEE Int. Conf. Image Processing*, vol. 4, pp. 2363–2366, Singapore, Oct. 2004.
- [8] R. Ulichney and M. Gaubatz, "Perceptual-based correction of photo red-eye," in *Proc. Int. Conf. Signal and Image Processing*, pp. 526–531, Honolulu, HI, Aug. 2005.
- [9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518, Kauai, HI, Dec. 2001.
- [10] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. ImageProcessing*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.
- [11] B. Li, Y. Qi, and X. Shen, "An image inpainting method," in *Proc IEEE Int. Conf. Computer Aided Design and Computer Graphics*, vol. 6, pp. 60–66, Hong Kong, China, Dec. 2005.
- [12] J. J. de Dios and N. Garcia, "Face detection based on a new color space YCgCr," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 902–912, Barcelona, Spain, Sep. 2003.
- [13] J. J. de Dios and N. Garcia, "Fast face segmentation in component color space," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 191–194, Singapore, Oct. 2004.
- [14] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, New York, McGraw-Hill, 1995.
- [15] J. E. Richman, K. G. McAndrew, D. Decker, and S. C. Mullaney, "An evaluation of pupil size standards used by police officers for detecting drug impairment," *Optometry*, vol. 75, no. 3, pp. 175–182, Mar. 2004.
- [16] D. K. Martin and B. A. Holden, "A new method for measuring the diameter of the in vivo human cornea," *Am. J. Optometry Physiological Optics*, vol. 59, no. 5, pp. 436–441, May 1982.
- [17] *Adobe Photoshop CS2*, San Jose, CA: Adobe, 2005.

- [18] *STOIK RedEye AutoFix 3.0*, Russia: STOIK Imaging, 2007.
- [19] R. Ulichney, M. Gaubatz, and JM Van Thong, "Redbot – a tool for improving red-eye correction", in *Proc. IS&T/SID Eleventh Color Imaging Conference: Color Science and Engineering System, Technologies, Applications*, Scottsdale, AZ, Nov. 2003.
- [20] F. Gasparini and R. Schettini, "Automatic red-eye removal for digital photography," in *Single-Sensor Imaging: Methods and Applications For Digital Cameras*, R. Lukac, Ed. CRC, 2008.