Journal of Imaging Science and Technology® 54(3): 030404–030404-7, 2010.

Automatic Redeye Correction Algorithm with Multilevel Eye Confirmation

Sony George
International School of Photonics, Cochin University of Science and Technology, Cochin, India

Jon Y. Hardeberg▲
The Norwegian Color Research Laboratory, Gjøvik University College, P.O. Box 191, N-2815 Gjøvik, Norway
E-mail: jon.hardeberg@hig.no

Tomson G. George
Wipro Technologies, Cochin, India

V. P. N. Nampoori
International School of Photonics, Cochin University of Science and Technology, Cochin, India

Abstract. *Redeye is a well-known problem in portrait photography. This effect is caused by the light entering the subject's eye through the pupil and reflection from the retina back to the imaging sensor. Many of the existing redeye correction methods sometimes fail to detect the actual redeye artifacts and incorrectly change other red colored areas. The proposed algorithm describes a fully automatic redeye correction system with multilevel eye confirmation stages. The algorithm first identifies the skin and locates the redeye region using color information. The detected region is then confirmed as redeye by its redness variation, glint, eye-lips triangle, and comparison with surrounding regions. The algorithm removes the falsely extracted components by verifying with rules derived from the spatial and geometrical relationships of facial components. Finally the defect is corrected by desaturating the red pixels while preserving the natural glint presence. © 2010 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2010.54.3.030404]

INTRODUCTION

Redeye is a common problem in portrait photography. This artifact is caused by the flash light which enters through the pupil and its reflection from the blood cells of the retina. The appearance of the redeye artifact depends on several factors, including the angle of the flash beam (the closer the outgoing beam is to the reflected beam, the greater the redeye effect), the age of the subject (the younger the subject the wider the pupil), and the ambient light (the lower the light the more the subject's pupil opens to admit more light). In particular because of the closeness of the flash to the optical axis and the high need for flash use because of the small lens aperture, redeye is a particularly serious problem in many modern and easy to carry consumer cameras. Figure 1 depicts a typical redeye artifact.

▲IS&T Member.

Received Dec. 3, 2009; accepted for publication Mar. 23, 2010; published online May 6, 2010.
1062-3701/2010/54(3)/030404/7/$20.00.

Several techniques are available in software and hardware form to reduce this problem. One of the widespread solutions is to adopt multiple flashes which make the pupil to be contracted before the actual photography is taken. Although it prevents the redeye artifact to some extent, it results in more power consumption and increased delay, which is not desirable.

Another way to reduce the problem of redeye is to use redeye correction algorithms included in software applications. Currently, many image processing software applications in the market offer redeye removal solutions. Redeye reduction software basically come in two forms; fully automatic algorithms where the redeye region is selected with no user input, using low level features such as color, shape, and semiautomatic ones in which the user has to specify eye region either click on the redeye or draw a box containing it before the redeye correction technique is applied.

Digital redeye correction algorithms can, in general, be divided into two stages; in the first stage redeyes are detected. In the second stage, the redeyes that were detected in the first stage are corrected. Detection commonly involves finding the redeye candidates from the face and then con-

Figure 1. Typical redeye effect.

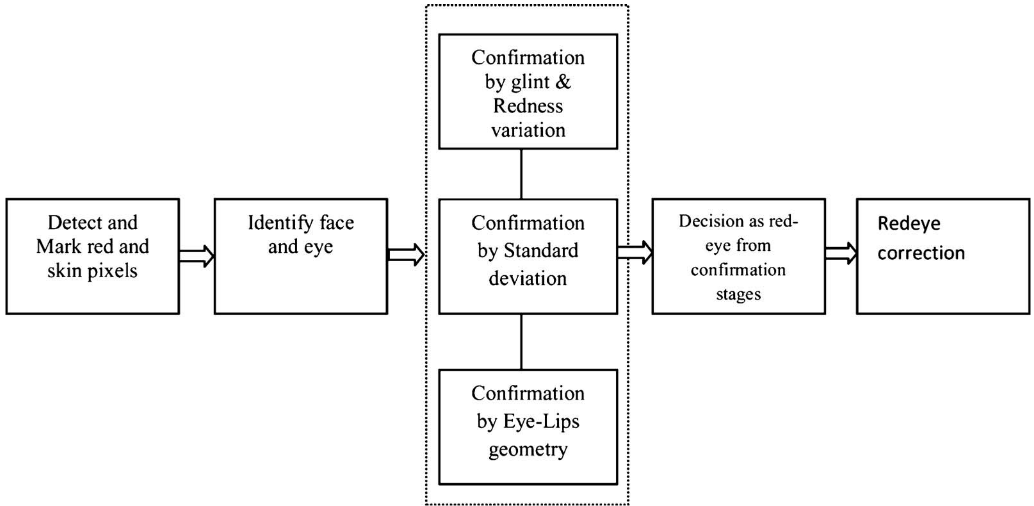

Figure 2. Block diagram of the main parts of the proposed redeye correction method.

firming that the detected part is an eye. A typical problem of most of these tools is poor pupil segmentation which leads to unnatural redeye correction and as a final product a more distorted image.

The main goal of the proposed algorithm is to accurately detect and remove redeyes from a picture, while avoiding false positives completely, which is the biggest problem of camera integrated algorithms or distributed software tools. At the same time, we want to keep the false negative rate as low as possible. Even though several automatic correction techniques have been proposed previously,[1–5] only few perform eye verification tests to confirm the located candidate as redeye.[6] It is worth mentioning here that using methods such as machine learning frameworks will increase the complexity and are computationally expensive. Our algorithm (Figure 2) starts with skin and red pixel extraction and proceeds by hybrid confirmation stages including confirmation by glint and redness variation, statistical properties of eye region and eye-lips geometry and finally the detected region is corrected to preserve natural appearance. Combining the three stages reduced the probability of false eye detection to a great extent. We use simple and computationally efficient confirmation techniques, which in fact can be implemented real time imaging applications.

After a review of the state of the art in the next section, we present in detail the different steps of our proposed algorithm, with particular focus on the confirmation stages. Then we present and discuss experimental results in which our methodology is confirmed on a large image database before we finally round off with a brief conclusion.

STATE OF THE ART

There are several publications and patents dealing with redeye correction. In this part of the paper we give an overview of various existing techniques which are either in software/hardware implemented for redeye correction. Most commercial image editing packages such as ADOBE PHOTOSHOP™, PAINTSHOP™, COREL DRAW™, or PHOTOPAINT™, offer the facility to correct redeye artifacts, and each of them has different degree of success. Most of them require some form of human intervention for the redeye detection and correction. Also some companies such as Canon, Kodak, Hewlett-Packard, Nikon, and Fuji have developed fully automatic redeye correction tools. A usual problem in most of these techniques is the poor segmentation which introduces new defects to the image.

To our knowledge, the first description of a method to correct redeye was given by Dobbs and Goodwin[1] of Eastman Kodak Co. In their work, the redeye is detected using the roundness property and the amount of red in a specific region selected by the user. Correction parameters of this algorithm were user controlled.

Later Benati et al.[2,7] modified the Kodak patent in which the region of interest in the image is defined by the user, followed by a thresholding operation in the HLS color space, in order to identify candidate pixels for correction. Then, the candidate pixels are grouped into one or more spatially contiguous groups. The group containing the redeye is then sought identified by a process where first all the pixels in the groups are given "scores" based on their color as well as on the shape of the group they belong to, then a region growth algorithm is applied to each group to improve. They also adapt a new correction technique which gave more natural appearance to the corrected eye. Lin et al.[3] of Hewlett-Packard also came up with an algorithm where the user only has to select a box around the eye.

The first fully automatic correction algorithm was to our knowledge developed by Schildkraut et al.[4] which is a modified version of the previous Kodak patent where they used eyes and face detection for automatic redeye detection. Afterwards many algorithms were introduced from various research groups as well as from industries. One such procedure was published by Wang and Zhang,[5] based on face detection and eye detection technologies such as neural network and principle component analysis. Gaubatz and Ulichney[8] introduced an algorithm with multiscale face classifiers. They define a measure of the redness as the ratio between energy of the red component and the energy of the remaining components. Later on modification of this work[9–11] resulted in a system called "RedBot." Luo et al.[12,13] introduced a two stage module where the redeye detection part is modeled as a feature based object detection problem, which uses the Adaboost algorithm[14] to simultaneously select relevant features and assign feature weight for the classifier. A systematic orientation independent feature computation scheme for object detection is adopted, which can detect redeyes of arbitrary in-plane rotation.

Hardeberg et al.[15–17] introduced in 2001 a semiautomatic method to correct the redeye based on an image mask which is computed by calculating a colorimetric distance between a prototypical reference redeye color and each pixel of the image containing this artifact. Morphological operations are applied to the binary mask, followed by a blob analysis method to group the pixels of the mask into eight-connected components. The blob that has the highest probability of representing a redeye artifact based on its size and shape is chosen. At the last stage the mask is smoothed, to achieve a softer correction that appears to be more natural to the human viewer. Smolka and collaborators[18] presented a computationally efficient algorithm where the redeye effect is detected using a skin detection module and the eye colors are restored using morphological image processing. This redeye correction system combined the manual as well as automatic detection modules.

Wan et al.[19] introduced techniques based on a statistical approach called active appearance models (AAM) which was able to detect even deformable object shape. Joining color information and a deformable model they locate redeyes as deformable objects. Approaches by Volken et al.[20] detect the eye itself by finding the appropriate colors and shapes without input from the user. They first look for the red zones of the whole image, then estimate the probability of each of these zones being a redeye by evaluating their roundness, the amount of white around these zones, corresponding to the sclera and the amount of skin. This technique was implemented in a web-based application[21] to allow people to correct their images online. Gasparini and Schettini[22] designed a modular procedure for automatic correction of redeye artifact in images of unknown origin, maintaining the natural appearance of the eye. Initially, a smart color balancing procedure is applied. This phase not only facilitates the subsequent steps of processing but also improves the overall appearance of the output image. Combining the results of a color-based face detector and of a face detector based on a multiresolution neural network the most likely facial regions are identified. Redeye is searched for only within these regions, seeking areas with high "redness" satisfying some geometric constraints.

Only few solutions have been developed specifically for the redeye confirmation stage. Marchesotti et al.[6] designed an adaptive system which selects the correction strategy based on three correction methods and compared according to their image degradation risk and their expected perceptual quality improvement. The latest development to our knowledge was from Yoo and Park[23] who proposed a redeye removal algorithm using inpainting and eye-metric information. In this algorithm face detection is performed initially, and then redeye regions are segmented using multicues such as redness, and color information. For redeye correction, pupils are painted with the appropriate radii calculated from the iris size and size ratio.

SKIN AND RED PIXEL CLASSIFICATION

Like for many other image processing algorithms related to human subjects, skin segmentation is the first stage of the proposed algorithm, and a subsequent process for face detection depends on the skin/nonskin classification. Due to the wide range of applications, skin detection has received considerable attention over the years. Research in skin detection has made use of many techniques, such as neural networks,[24] template matching, point distribution models, eigenfaces, Fisher faces and statistical approaches[25] such as support vector machines, higher order statistics,[26] etc. It has been observed that color modeling[27] is an efficient tool to detect skin pixels. In the skin detection application, the input is simply "skin" or "nonskin." The output can be any set of coordinates in RGB, HSV, or any other color space. Modeling skin color requires choosing an appropriate color space and identifying a cluster associated with skin color in this space. Various color space models are available and out of those the HSV color space has been shown to yield the lowest error rate for face detection.[28] Also the HSV color space confine the skin color to a more narrow region than RGB or YCbCr, therefore the segmentation can be done more precisely. The skin region can be extracted by converting to HSV domain and with a suitable threshold operation. One of the advantages of using color information as the key feature to detect and localize is that color is generally invariant to rotations, translations, and scale changes.

In the HSV cylindrical color space, H (hue) describes the shade of the color, S (saturation) express how pure the hue (color) is, while V (value) describes the brightness. There is a fairly distinct separation between skin and nonskin in the hue and saturation coordinates with less separation in the value coordinate. The removal of the V component takes care of varying lighting conditions. H varies from 0 to 1 on a circular scale, i.e., the colors represented by $H=0$ and $H=1$ are the same. S varies from 0 to 1, 1 representing 100% purity of the color. H and S scales are partitioned into 100 levels and a color histogram is formed using H and S.

Considering a pixel $X(i,j)=[R(i,j),G(i,j),B(i,j)]=[H(i,j),S(i,j),V(i,j)]$, it is classified as a redeye pixel belonging to the class R as

$$X(i,j) \subset \mathrm{R} \ \text{ iff } \ R(i,j)/G(i,j) > t_{RG} \ \text{ and } \ R > \min_{i,j} R(i,j), \tag{1}$$

where the threshold t_{RG} is set empirically to 1.8. Furthermore the pixel is classified as belonging to the skin region S as

Figure 3. Skin detection stage.

$$X(i,j) \subset S \text{ iff } V(i,j) > 0.4 \text{ and } S(i,j) \in [0.2,0.6] \text{ and } H(i,j) \in [0,25] \cup [355,360]. \quad (2)$$

While the resulting skin color segmentation was certainly a nice approximation, there were still some annoying artifacts. Using a region growing technique, the skin region and red region are grown independently. After the primary skin nonskin classification, morphological opening and closing applied over the area to remove unwanted "noise" pixels and to fill "holes." An opening by reconstruction operator is applied first on the binary image that is obtained after thresholding. This operation is nothing but erosion followed by dilation using a diamond structuring element (SE) with size four. Erosion removes small and thin isolated noiselike components. Dilation preserves those components that are not removed during erosion. Hence, the effect of using area open is removal of small and bright regions of the thresholded image. This step is followed by closing by reconstruction. Here, dilation followed by erosion with a diamond structuring element is performed. Initial dilation closes any small holes that may have been created during opening. Erosion removes the extra pixels that are added to the contour of the preserved components. Size of the SE set as the same during both opening and closing and should be greater than that of the smallest face the system is designed to detect.

Color information alone will not give correct classification in all cases. We can improve the skin detection by using edge information and modifying the way we classify skin pixels. We use a Sobel gradient to find the edges to classify pixel as skin, since the Sobel operator is fast, detects edges at finest scales, and provides smoothing along the edge direction, which avoids noisy edges. For a pixel to be classified as "skin" it should exceed the skin threshold for color histogram (H,S) as well as have a gradient less than a certain threshold called edge threshold. Figure 3 shows the output image after the skin detection stage. By analyzing the algorithm performance as a function of experimental output we identified the threshold values. Selection of an optimum threshold is important as it affects the later stages of the detection process. A lower threshold is better because further analysis is based on connected component operators. Increasing the threshold will increase the chances of losing certain skin regions. The skin threshold and edge threshold are set to 0.15 and 130, respectively. Each detected red areas is labeled and validate as redeye in the preceding stages of the algorithm.

Figure 4. Glint formed during photography.

REDEYE CONFIRMATION

One of the many challenges in redeye correction is to confirm the identified red region as a part of redeye. The redeye confirmation stage affirms each detected red region as being either part of an eye or not. The confirmation stage employs techniques which are computationally complex and most of them operate on blocks of pixels. The identified redeye regions can then be made available to other modules for subsequent correction. We propose a multilevel redeye confirmation algorithm, which facilitates minimization of the false detection. Three different confirmation approaches based on the presence of glint, statistical, and geometrical properties of eyes are used and combined, as described in the following sections.

Confirmation By Glint and Redness Variation

The first part of the confirmation algorithm verifies the redeye by ensuring the presence of glint: the unique property in eyes formed by the direct reflection of the light from the camera flash on the surface of the cornea. Since flash is the cause of redeye, reflection of the flash is often highly visible in the eyes of the subject (Figure 4). If a redeye is visible to a human observer, it is likely that the artifact is surrounded by pixels of a different degree of redness, creating a variation in the redness measure.[29] Redness is defined as the ratio between the energy of the red component "R" and the energy of the remaining green components "G" and blue "B." The Laplacian operator can easily locate such sharp changes in brightness. However, it will also emphasize pixels with excessive noise, which is undesirable. We have utilized a template of area 6×6 pixels to validate the presence of glint among red pixels.

As shown in Figure 5 for position G, the difference between certain pixels, G_1, G_2, G_3, G_4, and their neighbor pixels in a local 6×6 area is calculated in the red channel of the image as the following:

$$\mathrm{Var}(Gi) = \frac{1}{4}X\sum_{i=1,2,3,4} I(Gi) - \frac{1}{16}\sum_{B\in shadow} I(B). \quad (3)$$

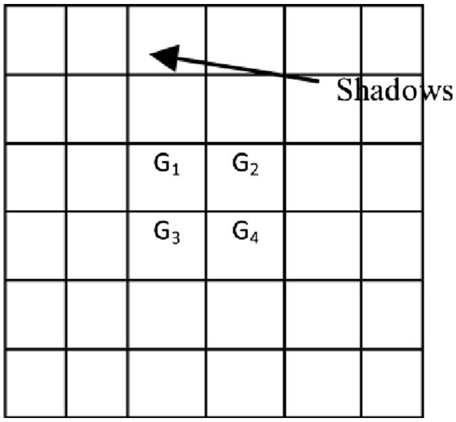

Figure 5. Template used for validating the presence of glint.

If $\mathrm{Var}(Gi) > 150$, and the number of red color pixels marked in skin and red pixel detection stage in this 6×6 area is greater than 10, then the pixel G will be considered as part of glint.

Confirmation By Standard Deviation

The second section of the algorithm performs confirmation check to red region detected in order to ensure that it is red enough itself and more red than its surroundings to satisfy the requirements to consider it as redeye. These verification tests are applied to candidate redeye areas. Candidate redeye pixel areas are filtered from the preliminary set based on redness contrast thresholds. Rectangular blocks are generated on the top, bottom, and the two sides of the red region. These four blocks have exactly same size, which is equal to the size of the red region. From experiments we observed that the one of possible region in face where the prominent change in color occurs in is the redeye region. Even though the color of the pupil becomes red during the flash, the surround would remain a white region. In addition to an eye is the lip region where there a further possibility for false detection. However, the change in color of the neighboring pixels will not be very prominent and this would not satisfy the above condition. Redness of the identified region (E_{red}) is compared with four surrounding blocks ($N_{\mathrm{red[i]}}$). Standard deviation (SD), of each rectangular block is calculated and if SD of the pixel content in each block is above a threshold TH_{red} then it is considered as a valid redeye region (Figure 6);

$$E_{\mathrm{red}} - \sum N_{\mathrm{red[i]}} > TH_{\mathrm{red}}. \tag{4}$$

Confirmation By Eye-Lips Geometrical Property

The last step of the confirmation algorithm is to validate the redeye using the geometrical locations of the mouth and eyes. This approach helps to directly locate eyes, mouth, and face boundary based on measurements derived from the colorspace components of an image. Since the YC_bC_r colorspace has been identified to be the best suited for classification of eye-lips,[30] we use this color space for this part of the algorithm. An analysis of the chrominance components indicated that high C_b and low C_r values are found around

Figure 6. Standard deviation stage.

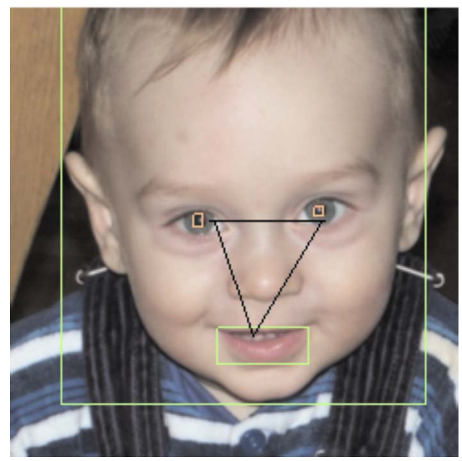

Figure 7. With eye lips detection.

the eyes. The eye map in the chroma is constructed from C_b, the inverse of C_r, and the ratio C_b/C_r. The two resulting eye maps are combined by a multiplication operation. The resultant eye map brightens both the eyes and suppresses other facial areas. The mouth region contains excess red component, compared to blue component, than other facial regions. Hence, the chrominance component C_r, is greater than C_b, near the mouth areas. We further notice that the mouth has a relatively lower response in the C_r/C_b feature but a high response in the C_r^2. Therefore, the difference between C_r^2 and C_r/C_b can emphasize the mouth regions. We form an eye-mouth triangle for all possible combinations of the two eye candidates and one mouth candidate. Each eye-mouth triangle is verified by checking (i) luma variations in eye and mouth blobs, (ii) geometry and orientation constraints of eyes-mouth triangles, and (iii) based on the locations of eye-mouth candidates, our algorithm first constructs a face boundary map from the luma. A triangle formed by the identified eye region and lips and if the height to width ratio falls within the range of well-known golden ratio tolerance (Figure 7).

From the analysis of redeye images it is observed that, in the majority of cases, redeye artifacts are generated when flash light falls at an acute angle, and hence it is less probable for this to occur for images that have been taken from the side of the subject. Most noise components can be removed after this stage. However, it might be still possible that some

Figure 8. Corrected redeye region by using 9 pixel averaging and 25 pixel averaging.

confusing combinations need to be dealt resulting in false detection with single eye images. To solve the ambiguity, we propose a method to assign equal weight for each stage and as the algorithm satisfies any of two criterions, the area would mark as redeye.

REDEYE CORRECTION

The ultimate aim of a redeye correction algorithm is to eliminate red pixels in the pupils, and at the same time preserving important details such as glint (i.e., white curvature reflections in the cornea) and borders between pupil and iris. Only few solutions[31,32] have been developed specifically for the correction step. Most approaches simply desaturate the pixels in the detected redeye regions.

We propose an algorithm, in which each pixel in the detected redeye region, is corrected by first calculating the average of nine surrounding pixels, and this reference is then processed as given in Eq. (5), below. From the confirmation step, the size of the red region is obtained and also the reddishness of the region. With reference to this data the alpha value is determined. Alpha value is calculated as the number fraction of red pixels included in each 5×5 pixels within the red block confirmed in the last stage;

$$R_{out}(i,j) = \alpha 0.05R_{in}(i,j) + 0.6G_{in}(i,j) + 0.3B_{in}(ij) + (1 - \alpha)R_{in}(i,j). \tag{5}$$

The correction is thus mainly performed by reducing the red content and by increasing the green and blue elements.

As shown in Figure 8, we checked the effect of the neighbor pixel averaging with nine and 25 pixels and could not observe any changes and so we chose nine pixels for averaging for simplicity.

RESULTS AND DISCUSSION

We have presented an efficient technique to correct redeye artifact in digital images. Our method first segments the skin and red regions based on color. The main emphasis of our approach is on the detection and confirmation part. We overcome the difficulty in ensuring that the detected region is redeye by using a hybrid algorithm where we used glint variation, standard deviation, and eye-lips geometrical properties of the eye region. In order to reduce the false positive detection rate we took images that satisfy at least two confirmation stages. These images are corrected in such a way that the red color is removed but the eyes maintain a natural look.

We have tested our algorithm over a database of 250 color images used in previous research by Smolka et al.[18]

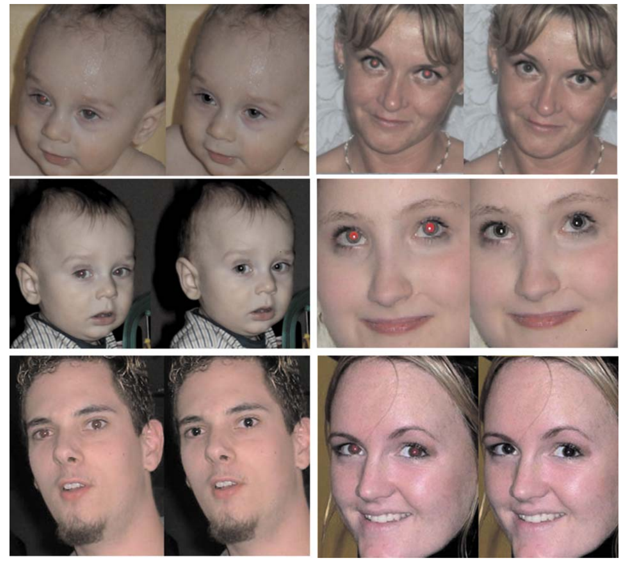

Figure 9. Examples of redeye correction results.

Figure 10. Examples of redeye correction results.

The images are of different resolutions, obtained with different digital cameras, under varying lighting conditions; all containing redeye artifacts. We have found that it provides visually very good results as exemplified in Figures 9 and 10. The proposed multilevel confirmation algorithm gives corrected image outputs for images with different sizes and a wide range of facial variations, position, and orientations. Further, the algorithm can detect both dark and bright skin tone under different lighting conditions. Figs. 9 and 10 show the results for subjects with some facial variations. In fact, the redeye correction is strongly related to the output of the detection module and this part is quite well performed by the aid of multilevel confirmation stages in the presented algorithm. Since there exists no publicly available reference database for redeye correction, the study performed cannot be quantitatively compared with the results reported by different authors in their papers.

CONCLUSIONS AND PERSPECTIVES

In this article an efficient redeye correction algorithm has been presented. The multilevel confirmation stages in the detection phase reduced the probability for false negatives.

There are also several opportunities for further research in this area, for instance, with respect to benchmarking. Also a reduction in computation and increase in speed could be achieved by reducing the number of stages for confirmation. Once a redeye has been confirmed in the first or second stage, the third confirmation step could for instance be skipped.

REFERENCES

[1] M. Dobbs and R. M. Goodwin, "Localized image recoloring using ellipsoid boundary function", US Patent 5,130,789 (1992).

[2] P. Benati, R. Gray, and P. Cosgrove, "Automated detection and correction of eye color defects due to flash illumination", US Patent 5,748,764 (1998).

[3] Q. Lin, D. R. Tretter, K. Kostantinides, and A. Patti, "Apparatus and a method for reducing redeye in a digital image", US Patent 6,016,354 (2000).

[4] J. Schildkraut and R. Gray, "Computer program product for redeye detection", US Patent 6,252,976 (2001).

[5] J. Wang and H. Zhang, "Apparatus and a method for automatically detecting and reducing redeye in a digital image", US Patent 6,278,491 (2001).

[6] L. Marchesotti, M. Bressan, and G. Csurka, "Safe redeye correction plug-in using adaptive methods", *Proc. IEEE Intl. Conf. on Image Analysis and Processing: Workshops* (IEEE Computer Society, Washington, DC, 2007) pp. 192–195.

[7] P. Benati, R. Gray, and P. Cosgrove, "Automated detection and correction of eye color defects due to flash illumination", US Patent 5,432,863 (1995).

[8] M. Gaubatz and R. Ulichney, "Automatic redeye detection and correction", *Proc. IEEE International Conference on Image Processing (ICIP)* (IEEE, Piscataway, NJ, 2002) pp. 804–807.

[9] R. Ulichney, M. Gaubatz, and J. M. Van Thong, "RedBot: A tool for improving redeye correction", *Proc. IS&T/SID Eleventh Color Imaging Conference* (IS&T, Springfield, VA, 2003).

[10] M. Gaubatz and R. Ulichney, "System and method for automatically detecting and correcting redeye", US Patent 7,155,058 (2006).

[11] *Redbot: Automatic redeye correction*, http://www.hpl.hp.com/research/redbot/ (accessed April 2010).

[12] H. Luo, J. Yen, and D. R. Tretter, "Detecting and correcting redeye in a digital image", US Patent 7,116,820 (2006).

[13] H. Luo, J. Yen, and D. R. Tretter, "Detecting and correcting redeye in an image", US Patent 7,333,653 (2008).

[14] R. E. Schapire, "A brief introduction to boosting", *Proc. 16th International Joint Conference on Artificial Intelligence*, Princeton, NJ (1999) pp. 1–6.

[15] J. Y. Hardeberg, "Redeye removal using digital color image processing", *Proc. IS&T PICS Conf.* (IS&T, Springfield, VA, 2001) pp. 283–287.

[16] J. Y. Hardeberg, "Digital redeye removal", J. Imaging Sci. Technol. **46**, 375–381 (2002).

[17] J. Y. Hardeberg, "Redeye removal using color image processing", US Patent 6,728,401 (2004).

[18] B. Smolka, K. Czubin, J. Y. Hardeberg, K. N. Plataniotis, M. Szczepanski, and K. Wojciechowski, "Towards automatic redeye effect removal", Pattern Recogn. Lett. **24**, 1767–1785 (2003).

[19] J. Wan, X. Ren, and G. Hu, "Automatic redeye detection based on AAM", *Proc. IEEE Intl. Conf on Systems, Man, and Cybernetics 2004*, Vol. **7** (IEEE, Piscataway, NJ, 2004) pp. 6337–6341.

[20] F. Volken, J. Terrier, and P. Vandewalle, "Automatic redeye removal based on sclera and skin tone detection", *Proc. IS&T Third European Conference on Color in Graphics, Imaging and Vision (CGIV)* (IS&T, Springfield, VA, 2006) pp. 359–364.

[21] *Redeye Remover*, http://ivrgwww.epfl.ch/software/red_eye_remover/RedEye Remover.html (accessed April 24,2010).

[22] F. Gasparini and R. Schettini, "Automatic redeye removal for smart enhancement of photos of unknown origin", Lect. Notes Comput. Sci. **3736**, 226–233 (2006).

[23] S. Yoo and R.-H. Park, "Redeye detection and correction using inpainting in digital photographs", IEEE Trans. Consum. Electron. **55**, 1006–1014 (2009).

[24] L. Chen and J. Zhou, "A skin detector based on neural network", *IEEE International Conference on Communications, Circuits and Systems and West Sino Expositions* (IEEE, Piscataway, NJ, 2002) pp. 615–619.

[25] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection", *Proc. IEEE Conf. Computer Vision and Pattern Recognition '99* (1999) pp. 274–280.

[26] N. Rajagopalan and K. Kumar, "Finding faces in photographs", *Proc. Sixth IEEE International Conference on Computer Vision* (IEEE, Piscataway, NJ, 1998) pp. 640–645.

[27] T. Yoo and S. Oh, "A fast algorithm for tracking human faces based on chromatic histograms", Pattern Recogn. Lett. **20**, 967–978 (1999).

[28] O. Ikeda, "Segmentation of faces in video footage using HSV color for face detection and image retrieval", *Proc. IEEE International Conference on Image Processing (ICIP)* (IEEE, Piscataway, NJ, 2003) pp. 913–916.

[29] L. Zhang, Y. Sun, M. Li, and H. Zhang, "Automated red-eye detection and correction in digital photographs", *Proc. IEEE International Conference on Image Processing (ICIP)* (IEEE, Piscataway, NJ, 2004) pp. 2363–2366.

[30] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images", IEEE Trans. Pattern Anal. Mach. Intell. **24**, 696–706 (2002).

[31] R. Ulichney and M. Gaubatz, "Perceptual-based correction of photo redeye", *Proc. 7th IASTED International Conference on Signal and Image Processing*, Honolulu, HI (2005).

[32] F. Chazli, F. Gasparini, and R. Schettini, "A modular procedure for automatic redeye correction in digital photos", Proc. SPIE **5293**, 139-147 (2004).