Model-Based Correction of Red Eye Defects

Andreas Held
Gretag Imaging AG
Regensdorf, Switzerland

Abstract

We present a fully automatic approach to the detection and correction of red-eye artifacts, as found in consumer photographs. The approach is model-based in that we first detect faces, then detect eyes in the found faces, and finally determine whether the eyes show any red-eye defects. Provided the detection stage found some red-eyes, we attempt to replace the red color by a suitable replacement color. Especial care is taken to ensure smooth transitions between corrected and original parts of the eyes.

1. Introduction

The automatic detection and correction of red eye defects is an important topic in photo finishing. Physically speaking, a red-eye defect occurs if a very strong light, a flash for instance, is reflected from the retina, that is, the back of the eye. The red color is indicative for the blood vessels in the retina. In terms of colorimetry, red-eye defects are not defined properly and there is a very thin borderline between defective and not defective eyes. In general, in red-defective eyes the values for the red channel will be considerably higher than the values for the two other channels.

In portrait images, the effect of red eyes is very disturbing and renders a photograph rather unpleasant if not unusable. In the past, several makers proposed and implemented methods for the assisted correction of red eyes, usually involving the user to actually point out the red eye to the system. This kind of user interaction is acceptable in a highly interactive work-flow with relatively little throughput. However, in an automated work-flow or a high-yield work-flow the operator interaction has to be kept to a minimum.

Academic literature about red-eye correction is sparse, only recently Jon Hardeberg described an interactive system as used by Conexant [2]. Besides, there are several patents by various makers and many more engineering approaches whose workings are, in general, not disclosed. What is common in all those approaches is the need for user interaction.

In the following, we will describe a system for the fully automatic detection and correction of red-eye defects in consumer photographs. Such systems are soon to be used in several products of Gretag Imaging.

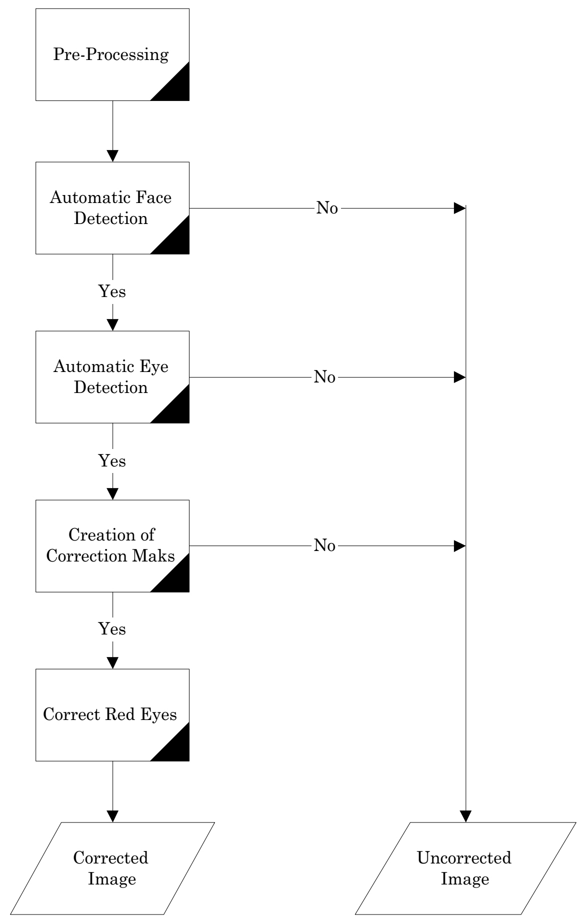

Figure 1: *Basic flow of the automatic red eye correction system*

2. System Overview

Our system for the automatic detection and correction of red eyes consists of the stages as shown in Fig. 1. There are several modules that, in turn, reduce the domain to be searched. There are two detection stages, automatic face detection and automatic eye detection, which are generalpurpose modules. This means, they do not necessarily use any knowledge about red eyes in particular. Therefore, those modules could extract any face from a photograph and any pairs of eyes or single eyes from a face.

The remaining two modules, the creation of red eye defect masks and the actual correction of the found defects are using knowledge from the domain of red eye removal, this is, they select and correct only those eyes which actually have red eye defects.

As can be seen from Fig. 1, images that are deemed to either not have a face, whose eyes can not be found, for whatever reason, or whose eyes do not appear to show any red eye defects are being left untouched.

3. Automatic Face Detection

For the actual detection of faces any system that fulfills this reasonably well will do. This could be for instance a neural network approach as proposed by Henry Rowley [3] or some wavelet based approach as proposed by Henry Schneiderman et al [4]. Of importance at this stage is that the detection of faces happens fully automatic, that the detection rate is reasonably high, and the false negative rate, that is, faces are being detected even though there is no face present, is reasonably low.

As most face detectors are not invariant to rotation we have to ensure that all the possible orientations of faces can be detected. How to do this will highly depend on the face detector being used, as the rotation invariance of each detector will vary widely. For instance, in Rowley's [3] approach, rotation invariance is given within approximately $\pm 15°$. On the other hand, in the approach by Henry Schneiderman [4], rotation invariance is given in a range of about $\pm 45°$. In both cases, real rotation invariance has to be insured by external means, this can, for instance, be done by pre-rotation of the image, followed by the normal face detection.

For a system based on the face detector by Henry Schneiderman, four stages are necessary. In other words, the face detector is applied to images rotated by $0°$, $90°$, $180°$, and$270°$, respectively

4. Automatic Eye Detection

Once a face has been detected, the search space for finding eyes can be restricted considerably. In general, we would obtain a bounding box of a face, together with its approximate orientation. As stated before, face detectors are in general not rotation invariant. Therefore, orientation of the face could be obtained in the range given by the rotational invariance of the face detector, which could be up to $\pm 45°$ in the case of the Schneiderman detector. Hence, the orientation as obtained from the face detector should be taken with a grain of salt but can still be used as general information. There are a variety of approaches that can be applied for detecting eyes. Again, similar to the detection of faces, it is important to have an approach that works fully automatic, has a high recognition rate, and a low false positive rate.

The basic approach for automatic eye detection is as follows. We perform a pre-processing, in order to enhance certain desired features of the image. This is followed by a first, coarse detection stage and by a second stage for refining the detection. Finally, we apply some heuristics to avoid unreasonable eye candidates.

As pre-processing step we could incorporate any processing that will enhance facial features, as for instance, histogram normalization, local contrast enhancement, or even red-enhancement for red-eye detection. In general, it is a good idea to normalize the input image, both in size and in lightness at this stage.

The actual eye detection stage follows the approach by Benn et al [1]. They propose a very interesting approach for the detection of eye centers based on the Hough transform. Although Hough transforms might not appear the best choice due to rather large requirements on memory and processing speed, they show that this need can be greatly reduced by using the so-called gradient-decomposed Hough transform. We adopted this idea and extended it to a multi-stage eye detector.

In the first stage, we calculate the gradients using a Sobel operator and perform Hough voting on the two gradient images. Smoothing the accumulator space and extracting maxima gives a first set of rough eye candidates. In a second stage, we extract a local neighborhood around the found eye candidates and perform the Hough transform within this neighborhood only. Different to the first stage, we directly calculate the projections of the Hough accumulator space in x and y direction, in general, this will roughly correspond to two Gaussian distributions. Mean and variance of the Gaussians can now be used as estimates for the position and the size of the eye. In addition, the number of votes the found position received during the first stage can be used as a relative confidence measure for the found eye.

Finally, during post-processing we can try to eliminate eye-candidates that are not plausible. This can for instance be done by taking into account some confidence measure as can be obtained from some eye detectors. By sorting the eye candidates according to their confidence and by further analyzing pairs of eye candidates according to distance and orientation in respect to the face, we are able to disregard those candidates that would yield highly unlikely constellations. In the end, we will keep at most two eye candidates per face. If there are two candidates, then they will fit a very coarse model of a face, as shown in Fig. 2.

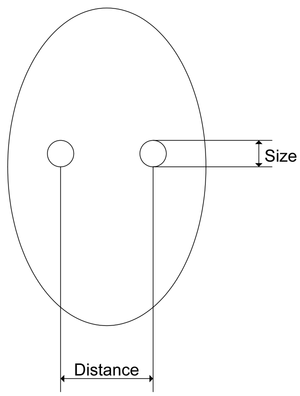

Figure 2: *Face Model*

5. Correction Mask

Once the centers of the eyes have been found we have to decide whether this eye has a red eye defect, and, if yes, what portions of the eye are defective. This task is commonly referred to as the creation of the correction mask. The correction mask has to actually capture two important aspects at once: the color of the actual defect and its geometry.

5.1. Red-Enhance Images

In order to reliably find eyes with red color defects we first convert the image into a red-enhanced space, which consists of only one color plane. The definition of the red-enhanced space is as given in Eq. (1).

$$I_{red} = R - \min(G, B) \tag{1}$$

where R refers to the red channel, G to the green channel, and B to the blue channel of the input image, respectively. This definition does result in red-eye defects being amplified and thus easily detectable. At the same time, it allows us to fulfill the first constraint for the creation of a correction mask, its spectral homogeneity.

5.2. Placing of Seed Pixels

According to the initial estimate for position and size of the iris, as obtained form the eye detector, we can decide whether there is a red defect in this eye and where exactly the strongest defect is located. This is shown in Fig. 3.

Whether a seed pixel really belongs to a red-eye defect is determined according to its HSV value and a fuzzy membership function. If we get the fuzzy intersection r of the three channels, see Eq. (2), then we can decide with a simple thresholding operation whether the seed belongs to the class of red-eye defects or not.

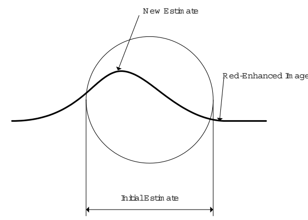

Figure 3: *Red Eye Model*

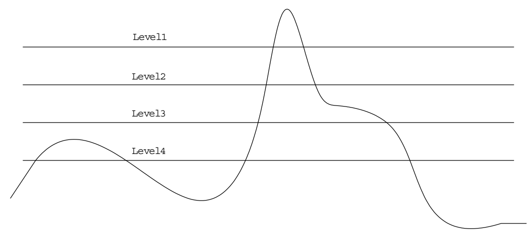

Figure 4: *Level Sets*

$$r = \frac{hsv}{\max(h, s, v)} \tag{2}$$

5.3. Level Sets

Based on the selected seed pixel we can now analyze the immediate neighborhood of this pixel. Our approach is based on the idea of pseudo level sets, or progressive thresholding. Analyzing the gray value distribution around the seed pixel we can find an optimal threshold for separating foreground from background pixels. Equally spacing between this optimal threshold and the maximum value of the seed pixel and applying a threshold at each of these computed values gives a progressive set of binary masks (pyramid), starting from a simple foreground/background separation up to the separation of the seed pixel. Careful analysis of the obtained masks from top to bottom allows to find the very mask having biggest size but still fulfilling a set of constraints about its circularity. This is shown schematically in Fig. 4. As this whole procedure is applied in the red enhanced space, the resulting mask fulfills certain homogeneity constraints implicitly.

5.4. Post-Processing

Several steps are performed at this stage in order to clean the resulting mask and to adapt it for further correction of the original image. In order to remove small holes and intrusions we perform a binary closing with a $7x7$ approximation to the disk structuring-element. Further, to remove small outgrows, we perform a binary opening with a $3x3$

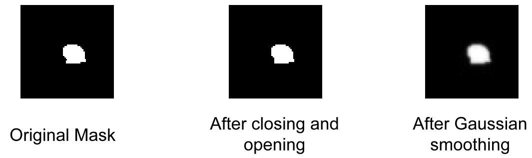

Figure 5: *Correction Masks*

structuring-element. Finally, we perform a binary dilation with a $5x5$ structuring element and do a Gaussian smoothing with a $7x7$ kernel. The effect of these operations is shown in Fig. 5.

Smoothing the eye-defect correction mask allows for a very effective correction in the next stage. As we are now not dealing with a binary mask, but rather with a gray-scale mask, it is relatively easy to obtain gradual corrections towards the borders of the red-eye defects. This, in general, looks far more natural than any sudden change.

6. Correction of Eye Defects

At this stage we have the uncorrected input image and a correction mask as outlined in the previous section. The correction mask is not a binary mask but rather a gray-level mask which is at the same time a measure for the probability that a certain pixel belongs to a red-defect region or not. Pixels along the borderlines receive a gradually decreasing probability, allowing for a smooth change between corrected and uncorrected regions.

If we assume the mask to show the actual probability values for eye defects in the range $[0...1]$, then we can state the correction for the defects as shown in Eq. (3).

$$R_{new} = R - m(R - \min(G, B)) \tag{3}$$

In words, if the probability of an eye defect is 0 then the correction factor is 0 as well. Otherwise, the red channel will be pulled towards the minimum of both, the green and blue channels. The effect of this correction is illustrated in Figures 6 and 7.

In case where the difference between the green and blue channel is rather large, we will have to adjust the bigger of the two as well, avoiding a rather unpleasant color shift. This adjustment is shown in Eqs. (4) and (5).

$$\text{if } G > R_{new} \text{ then } G_{new} = (R_{new} + B)/2 \tag{4}$$

$$\text{if } B > R_{new} \text{ then } B_{new} = (R_{new} + G)/2 \tag{5}$$

In the case where $m = 1$, this reduces to bringing every color onto the neutral axes, as in that case:

$$R = G = B = min(R, G, B) \tag{6}$$

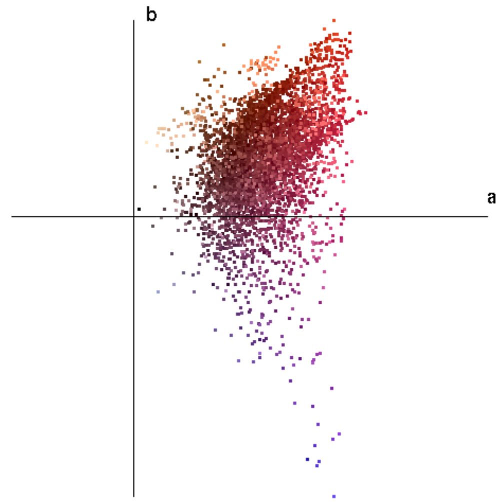

Figure 6: *Distribution of Red Eyes*

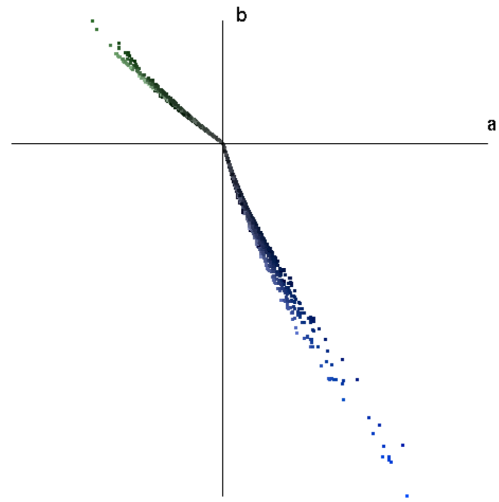

Figure 7: *Distribution of Corrected Red Eyes: After Step 1*

Images	Face Det.		Eye Det.		Red Eyes	
#	Corr	False	Corr	False	Corr	False
325	76.0	11.0	59.6	21.6	51.6	17.8

Table 1: *Statistical Detection Results [%]*

The net effect of the whole correction step is to replace the red defect with a neutral (or nearly neutral) color with minimum lightness, in effect leading to a rather darkish iris of an undefined color.

7. Conclusions

Detecting and correcting red-eyes in a consumer photograph is a challenging and difficult task. The main problems lie in that we have to avoid, by any means, the correction of so-called false positives, while maintaining a fair detection and correction rate. To address the first point, we proposed a hierarchical approach that consecutively reduces the search space by taking into account semantical information about the image in question. We believe that only such a relatively "high-level" approach will be able to perform well in the given settings. Secondly, to be usable in a commercial system, performance of the system must be very good. We achieve this again by the hierarchical approach, which allows us, at a relatively early stage, to abandon the processing of images that do not contain the necessary context. The actual workings of our approach are shown in Figs. 8 and 9.

The system has been tested on a variety of input images, a statistical overview is given in Table 1, whereas some actual processing results are shown in Fig. 10. In the latter figure, original eye regions are shown on the left, whereas the processed eye is shown on the right. It is especially interesting to see how highlights are preserved using the simple color processing as outlined in section 6. Still, in some cases a slight halo effect remains after processing. This will be addressed in a future version of the system.

The main problems, at present, are the eye detection step which does require an iris of sufficient size and clarity. Once artifacts of the image disturb the circular features of the eye and the eye region, then detection using the described circular Hough transform becomes instable, leading to a number of false positives. This has a particular effect on the false detection rate of the whole system, as shown in Table 1. However, preliminary work using an improved eye detection scheme indicates that the false detection rate of the overall approach can be reduced to less than 3%, with a slightly better detection rate. Work is now in progress on incorporating these findings into the system.

Figure 8: *Sample Input Image*

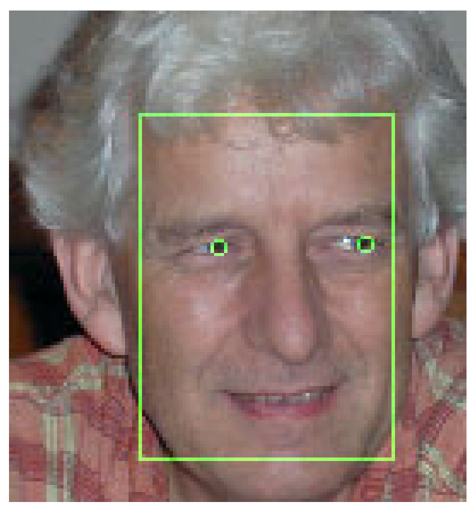

Figure 9: *Detection Results Overlaid*

Figure 10: *Results of red eye processing, see text for explanations.*

References

[1] David E. Benn, Mark S. Nixon, John N. Carter. ”Robust Eye Centre Extraction Using the Hough Transform.” Proc. 1st Int Conf AVBPA, pp 3–9, Crans-Montana, (1997).

[2] Jon Y. Hardeberg. “Red Eye Removal using Digital Color Image Processing.” Proc. PICS 2001, pp 283–287, Montreal, (2001).

[3] Henry A. Rowley. ”Neural Network-Based Face Detection.” PhD Thesis CMU-CS-99-117, Carnegie Mellon University, Pittsburgh, (1999).

[4] Henry Schneiderman, Takeo Kanade. ”A Statistical Method for 3D Object Detection Applied to Faces and Cars.” Proc. CVPR 2000, Vol I, pp 746–752 , Hilton Head Island, 2000.

Biography

Andreas Held received his BS in Electrical Engineering from Burgdorf Polytechnic, Switzerland, and his MS and PhD from Shizuoka University, Japan. From 1997 to 1999 he was with RWCP in Tsukuba, Japan, working on autonomous robots. At present, he is working at Gretag Imaging, Switzerland, specializing on color image processing and pattern recognition for consumer images. His special interests are in object detection and color reproduction. He is a member of IEEE and IPSJ.