Automatic red eye detection

Ilia V. Safonov
Samsung Research Center,
Moscow, Russia
Ilia . Safonov @ samsung . com

Abstract

The red eye artifacts are troublesome problem of consumer photos. Correction of red eyes without user intervention is an important task.

The novel method of automatic detection of red eyes is proposed. This approach is based on application of color information via 3D tables and edge information via directional edge detection filters. For classification a cascade of classifiers including AdaBoost classifier is applied.

The quality criterion of automatic red eye detection algorithms are discussed. Finally experimental results demonstrate high accuracy and efficiency of the proposed method in comparison with existing solutions.

Keywords: *Redeye detection, classification, boosting, directional edge detection filters.*

1. INTRODUCTION

A lot of consumer photos are influenced by red eye defect, which frequently arises when shooting with flash. This defect considerably worsens impression from the photo. This phenomenon caused in part by a small angle between the flash and the lens of the camera. This angle has decreased with miniaturization of cameras and cameraphones.

Several attempts have been made to reduce this problem. For example using one or several pre-flashes to reduce the size of human's pupils. Despite these efforts red-eyes still are a huge problem in amateur photography. Figure 1 demonstrates typical red eyes effect.

Figure 1: Typical red eye effect.

In whole the severity of the problem varies for different nations. The origin of red eye and factors affecting its severity were deeply investigated by the Kodak researchers [1]. When light impinges on the human eye at an angle not too far from the optical axis, it may propagate through the pupil, be reflected back from the fundus, and exit the pupil at approximately the same angle it entered. The fine blood vessels in the fundus color the reflected light red. The human pupil contains the pigment melanin, which gives it its dark color. The amount of melanin in the pupil correlates with skin and hair pigmentation, so that darker-skinned and darker-haired individuals typically have high melanin concentration in their pupils. Melanin attenuates the light propagating through pupil and higher melanin content leads to reduced red eye severity. Although there are some variations of pigmentation within races, it is still the case that a similar statement can be made for races, namely, that darker complexioned races on average exhibit less red eyes. However for Caucasian race (Europeans and white Americans) the red eyes effect is a big and troublesome problem. Table 1 contains results of experiments for different demographic groups in identical shooting conditions [1].

TABLE 1
PERCENTAGE OF OCCURRENCE OF VISUALLY DETECTABLE RED EYE FOR DIFFERENT DEMOGRAPHIC GROUPS OF VARYING RACE AND AGE

Group	*Detectable Red eyes, %*
African - American	9
Asian	15
Latin American	41
Caucasian Adult	70
Caucasian Youth	82

In present time general trend is detection and correction of red eyes on digital image with minimal user interactions or fully automatic. However majority of existing automatic solutions have serious drawbacks and limitations. Several algorithms are face orientation dependent; frequently they have high level of False Negatives (FN) and/or False Positives (FP) errors on detection stage; some solutions have high computational complexity and/or memory requirements. Thus, the development of automatic red eyes correction method with good detection quality, independent from face orientation, with relatively low computational complexity and memory requirements is an essential task.

2. ANALYSIS OF PHOTOS WITH RED EYES

For revealing problems in detecting red eyes we have collected and analyzed 900 photos. Photos were taken from private collections and Internet. Photos have various sizes and were taken by different photographers using various models of DSCs. Several scanned photos are also included into collection. We randomly divided collection on training set containing 850 photos and testing set containing 50 photos. Total number of red eyes in training set is 2738 (1304 paired and 130 single eyes).

By analyzing photos we selected the following main problems in automatic red eye detection:

- Color tone of red eyes varies significantly: from yellow and orange to magenta and violet;

- Color tone of red eyes and skin tone partially coincide;

- Face orientations vary widely and sometimes only part of the face is visible;

- Several red eyes have no pair.

We manually measured 5000 samples of skin tones and 5000 samples of red eyes tones. Figure 2 is a distribution of red eye tones and skin tones in a*b* plane of L*a*b* color space.

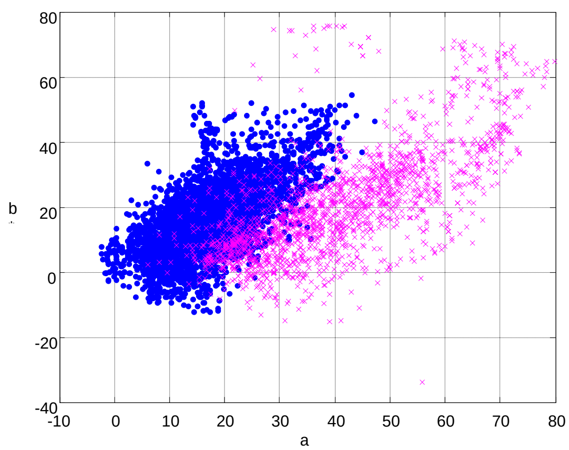


Figure 2: Projection of distribution of red eye tones (magenta crosses) and skin tones (blue points) to the a*b* plane.

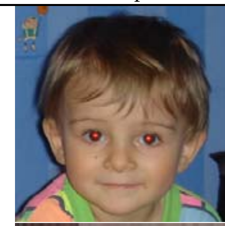
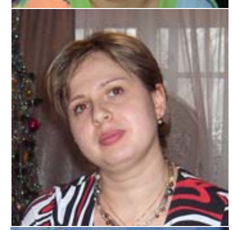
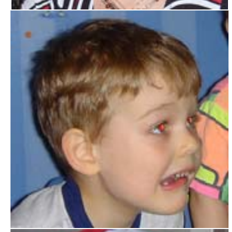
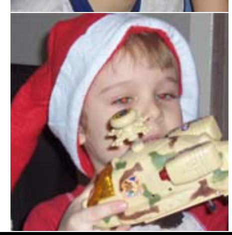
Possible colors of the red eyes lie in the wide range. Also there is an intersection between eye tones and skin tones. This effect is caused by mixture of the white glint (specular reflection on pupil) color and the red color of the eye retina. Brightness also doesn't allow to distinguish these tones fully.

At present time a lot of automatic approaches involve detecting faces in an image and, subsequently, detecting eyes within each detected face. But most of the face detection algorithms are only able to detect faces that are oriented in upright frontal view; these approaches can not detect faces that are rotated in-plane or out-of-plane with respect to the image plane, also can not detect faces in case when only part of face is visible. We counted percentage of face orientations on photos from our collection (see Table 2). In our opinion there are too many cases when face detection is not applicable.

All eyes can be divided into two groups: relatively large and well distinguishable and small with low local contrast. Let us name the first group HQ red eyes and the second group LQ red eyes. In training set 72% of red eyes is HQ and 28% is LQ. It is obvious that severity for HQ and LQ red eyes is different. It is necessary to correct majority of HQ red eyes, whereas correction of LQ red eyes is desirable.

Also we count number of red eyes on each photo. Figure 3 demonstrates distribution of number of red eyes on photo. Less than 3 % of cases of a photo contain more 6 red eyes.

TABLE 2
PERCENTAGE OF FACE ORIENTATIONS

Orientation	*Example*	*%*
Upright frontal		72
Rotated in-plane		17
Rotated out-of-plane		6
Only part of the face is visible		5

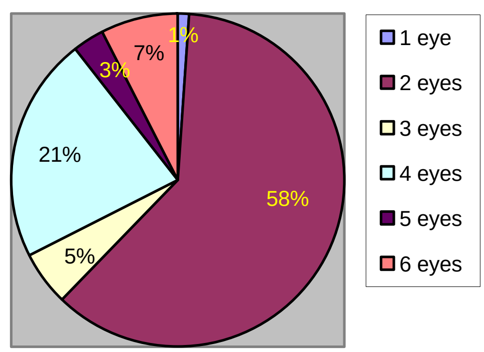


Figure 3: Percentage of the number of red eyes on photo.

3. RELATED WORK

There are many papers and patents dealing with semi-automatic and automatic red eye correction. All existing approaches include at least 2 phases: red eye detection and correction (retouching). Detection phase can consist of the following stages: human face or/and eye detection, segmentation and labeling of red regions, calculation of features and classification of regions, glint detection, pair verification.

Segmentation of red regions is performed usually using thresholding in some of the color spaces: YCbCr [2, 3], Lab [4], RGB [5]. Often, a special so-called redness factor is calculated for each pixel which is then used to make thresholding [6, 7, 8]. Two redness factors are applied simultaneously in [9] to increase the number of detected red eyes. However, due to the reasons mentioned above (see Figure 2) it's impossible to discriminate accurately between different red eye pixels and skin pixels using thresholding in any color space or thresholding by redness factor. That is why it is impossible to detect all types of red eyes via thresholding.

Technique described in [10] provides significantly better segmentation results. The red eye colors are bounded by curves in a*b* planes of L*a*b* color space. Parameters of curve depend on brightness (L) value. This approach does not allow to segment red eyes which have color typical for skin tones but such eyes occur rather seldom. The main disadvantage is expensive conversion to Lab color space.

Another group of algorithms apply for segmentation of red eye regions correlation or matching filters. The series of 24 symmetrical annular edge detection filters of increasing radius are applied to redness image in [11, 12]. However several red eye regions have non round form and applying of 24 convolutions is too expensive.

Luo's algorithm [13, 14] demonstrates combining of these two common approaches: thresholding of redness and matching filtration of redness image. Preferred redness image is a* component of L*a*b* color space. Filter produces final redness score as:

$$RS = AR_1 + w \times \left(\frac{AR_1}{AR_2}\right)^n , \qquad (1)$$

where AR_1 and AR_2 are average redness of inner and outer squares respectively according to notations from Figure 4, w is a weight factor, n is predefined constant. The first term here represents the absolute redness value and the second term represents the redness contrast between the inner square and its surrounding margins. Applying of this filter to redness image followed by a thresholding produces a binary image. To detect red eyes of different sizes filters of multiple kernel sizes are used and output binary images are combined by logical OR.

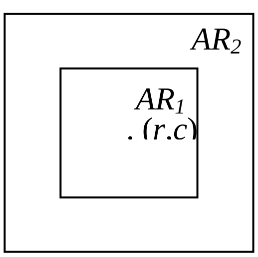


Figure 4: Concentric template of HP redness score filter.

It's obviously that after segmentation a lot of False Positives is detected. There are several common approaches which are used to decrease FP. Researchers form Microsoft [5] apply glint detection step and examine only red regions surrounding highlight pixels. However about 10-15% of red eyes have not glint.

Other popular approach is applying face detection before segmentation [3, 7, 8, 15, 16]. Frequently modifications of Viola-Jones algorithm [17] or detection of oval skin tone areas are applied. Face detection algorithms are orientation dependent and its applicability is limited as mentioned above. Sometimes jointly with face detection [3] or separately [13, 14] eye outline detection is used. Unfortunately these eye outline detection algorithms are orientation dependent too. Thus all these approaches increase FN for big sets of photos contained various types of red eyes in arbitrary orientation.

Paper [4] describes the following main specific characteristics of red eye regions:

- a reddish tone;

- a round shape;

- an area that is relatively small with respect to the whole image area;

- compactness;

- they are located in the neighborhood of skin patches;

- the presence of whitish pixels, associated with the sclera of the eye around the cornea, in close proximity to the region;

- the existence of relatively high local variations in contrast across the region.

In some way all red eye detection algorithms use these specific characteristics. The careful tuning of parameters has allowed Fotonation to create the decision tree for effective rejecting of FP [4].

General trend is to use more complicated machine learning techniques instead of decision tree. Boosting [18, 19] is a family of learning algorithms that produce classifier which is a weighted combination of weak classifiers. In [3, 10, 13, 14] the weak classifier is simple comparison feature value with threshold. Before AdaBoost classifiers committee, cascade of several classifiers for elimination of obvious FP is applied usually.

In addition algorithms from AdaBoost family have ability to select relevant and more informative features during training. In [13], AdaBoost training procedure is used for selecting 25 final features from initial features set containing 178 features.

Quality of any learning algorithm seriously depends from quality of training set. HP [20] realized automatic red eye reduction algorithm as free web service (www.redbot.net). The HP researchers motivations are collecting of plenty "real-world" typical consumer photos damaged by red eye effect and getting feedback from users about correction performance and quality. In our opinion this is really good way for fine tuning of the algorithm.

One more method used to eliminate FP is pair verification [14, 21]. In addition situation when only one eye from pair is corrected is unwanted especially for embedded implementations. However about 5% of photos have single red eyes, this happens, for instance, when face is partially screened. Thus pair verification is useful procedure but single eyes should not be ignored.

Obvious way for preventing processing of photos taken without flash is analysis of EXIF Flash tag [22, 23]. This approach is certainly useful to decrease processing time and reduce FP.

4. RED EYE DETECTION

4.1 Segmentation

Taking into account analysis outcomes and prior art information we propose to use color and edge information simultaneously for segmentation of red eyes.

In contrast to other red eye detection methods the proposed method does not use thresholding in any color space or in redness image, but uses multitudes of colors typical to red eyes and human skin defined by two 3D tables. Notice that these multitudes can intersect each other. Such tables can be constructed for each color space and can circumscribe 3D multitudes of arbitrary form. The majority of photos are stored in JPEG file format because processing is performed in YCbCr color space for preventing excessive color space conversions but in general proposed method can be adopted for any color space easily.

The 3D tables contain typicalness levels that characterize given color as a color of human skin and red eyes correspondingly. Typicalness level is an integer value ranging from 0 to 3, where 3 means that given color is very typical for human skin/red eyes; and 0 means that given color is not typical for human skin/red eyes. For setting typicalness level we analyzed distribution of 5000 manually labeled skin pixels and 5000 red eye pixels and have taken into account theoretically possible values of skin and red eyes colors. In certain sense, typicalness level is similar to posterior probabilities but we have made a lot of changes manually on initial stage and automatically on final stage by maximization of detection quality criterion (see bellow 17) for photos from the training set.

Notice that multitudes of colors typical for red eyes and human skin may be defined by a set of analytical functions, and these multitudes can intersect each other. Example of the applicable functions for skin tones is presented in [26].

Figure 5: *Color marks* for photo from Figure 1.

For each pixel of image the special *color mark* is set if skin typicalness is equal to 0 and red eye typicalness is greater than 1. White points in Figure 5 show *color marks* for photo from Fig. 1.

In contrast to other red eye detection methods proposed method does not use symmetrical matching or correlation filters, but use directional edge detection filters. It allows detecting regions of red eyes with different sizes without application of several filters with kernels of different sizes and allows detecting regions of red eyes of any form (not just round).

Figure 6: Redness of photo from Figure 1.

Course gradient filters, described for example in [27] can be applied as directional edge detecting filters for images. However we revealed that filters based on ratio instead of difference are more stable to noise and compression artifacts. The filtrations are applied to the redness image. The plenty of our experiments has shown that good estimation of redness is linear combination of brightness and red intensity. For example, for YCbCr color space the preferred implementation is Y+5Cr. Figure 6 demonstrates the redness of photo from Figure 1. We propose the following four directional edge detection filters (see Figure 7):

$$A_0(r,c) = \sum_{i=r-1}^{r+1} \sum_{j=c-1}^{c+1} (\mathrm{Y}(i,j) + 5 \times \mathrm{Cr}(i,j)), \tag{2}$$

$$A_1(r,c) = \sum_{i=r-4}^{r-2} \sum_{j=c-1}^{c+1} (\mathrm{Y}(i,j) + 5 \times \mathrm{Cr}(i,j)), \tag{3}$$

$$A_2(r,c) = \sum_{i=r+2}^{r+4} \sum_{j=c-1}^{c+1} (\mathrm{Y}(i,j) + 5 \times \mathrm{Cr}(i,j)), \tag{4}$$

$$A_3(r,c) = \sum_{i=r-1}^{r+1} \sum_{j=c-4}^{c-2} (\mathrm{Y}(i,j) + 5 \times \mathrm{Cr}(i,j)), \tag{5}$$

$$A_4(r,c) = \sum_{i=r-1}^{r+1} \sum_{j=c+2}^{c+4} (\mathrm{Y}(i,j) + 5 \times \mathrm{Cr}(i,j)), \tag{6}$$

$$E_1(r,c) = \frac{A_0(r,c)}{1 + A_1(r,c)}, \tag{7}$$

$$E_2(r,c) = \frac{A_0(r,c)}{1 + A_2(r,c)}, \tag{8}$$

$$E_3(r,c) = \frac{A_0(r,c)}{1 + A_3(r,c)}, \tag{9}$$

$$E_4(r,c) = \frac{A_0(r,c)}{1 + A_4(r,c)}, \tag{10}$$

where filtration is carried out for all pixels of the image, r is index of row, c is index of column, Y and Cr are corresponding channels of the image. Unity in a denominator is added to avoid singularity.

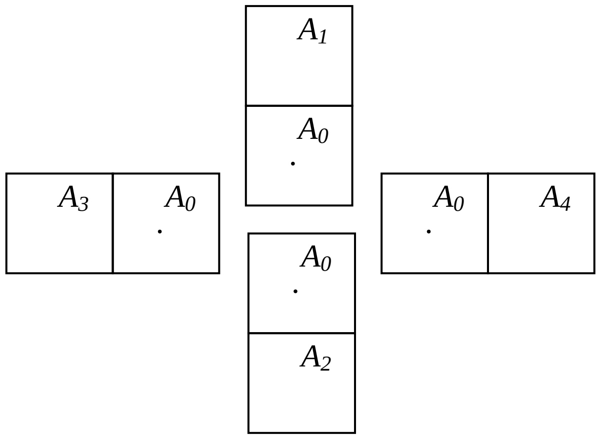


Figure 7: Four directional edge detection filters.

The computational complexity of this approach is identical to 5 convolutions with 3 by 3 kernels. For each pixel of image the special *edge mark* is set if maximal result of filtrations by $E_1(r,c)$ or $E_2(r,c)$ or $E_3(r,c)$ or $E_4(r,c)$ at this pixel is greater then a threshold value *T*, also index of the filter, which produces maximal result, is stored. Threshold *T* is not constant for all pixels of the image, it is increasing when color of the image pixel is typical for human skin color and decreasing when, color of the pixel is typical for red eye. All filter coefficients and thresholds were chosen by maximization of detection quality criterion (see bellow 17) for photos from the training set.

This approach allows detecting pixels of a red eye, whose color is close to color of skin tones and pixels of so-called "gold eyes". The described approach also allows to calculate informative features for the subsequent classification of regions.

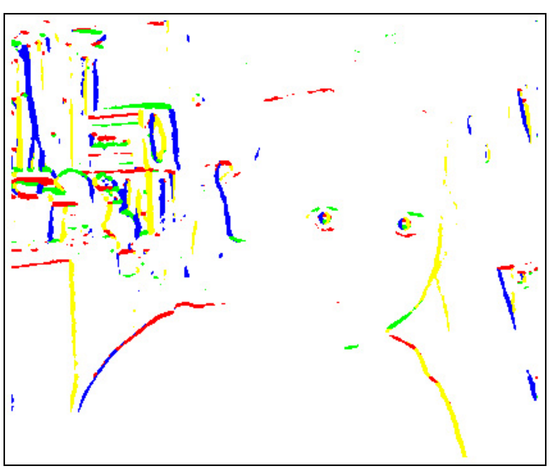


Figure 8: *Edge marks* for photo from Figure 1.

Figure 8 demonstrates *edge marks* for photo from Figure 1. The pixels marked by E_1 (7) are colored in red, marked by E_2 (8) are colored in green, marked by E_3 (9) are colored in blue, marked by E_4 (10) are colored in yellow. Notice that regions of red eyes generate the specific patterns which differ from other marked regions.

The results of these two segmentations, i.e. *color and edge marks*, are combined by OR logical operation before labeling. The proposed segmentation approach provides excellent detection result. For training set more than 99.9% HQ and about 97.5% of LQ red eyes are detected average number FP per image is 1897. The huge number of FP will be reduced drastically on classification stage.

4.2 Classification

The aim of classification stage is elimination of FP. The application of AdaBoost [18, 19] for both feature selection and classification is valuable. AdaBoost is a family of learning algorithms that produce classifier named as classifiers committee that is a weighted combination of weak classifiers. The weak classifier is any classifier that can achieve slightly better error rate than random guessing. AdaBoost has demonstrated three main advantages: good generalization capabilities, low computational complexity of a final committee and implementation simplicity. Nevertheless, Boosting turned out to overfit, if let run for too many training rounds, especially in case of noisy training data. The general scheme operation of classifiers committee is shown on Figure 9.

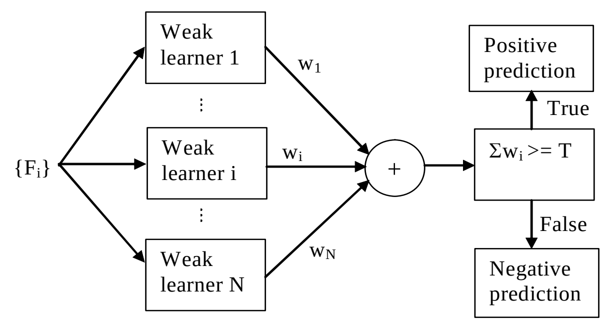


Figure 9: Scheme of AdaBoost classifiers committee.

There are several AdaBoost algorithms which differ by approaches for optimization of weights w_i. Some realizations of these algorithms have possibilities to adjust parameters of simple weak learners, in particular to optimize thresholds. We used GML AdaBoost Matlab Toolbox for feature selection, building of classifiers committee and adjusting parameters of weak learners. GML AdaBoost Matlab Toolbox is set of Matlab functions and classes implementing a family of classification algorithms, known as Boosting. Real AdaBoost, Gentle AdaBoost and Modest AdaBoost. Real AdaBoost is the generalization of a basic AdaBoost algorithm first introduced by Fruend and Schapire [18]. Real AdaBoost should be treated as a basic fundamental boosting algorithm. Gentle AdaBoost [28] is a more robust and stable version of real AdaBoost. As rule Gentle AdaBoost performs slightly better then Real AdaBoost on regular data, but is considerably better on noisy data, and much more resistant to outliers. Modest AdaBoost [29] is regularized tradeoff of AdaBoost, mostly aimed for better generalization capability and resistance to overfitting for certain specific sets of training data.

GML AdaBoost Toolbox supports weak learners as Classification and Regression Trees (CART). CART is a tree graph, with leaves representing the classification result and nodes representing some predicate. Branches of the tree are marked true or false.

Classification process in case of decision tree is a process of tree traverse.

For effective learning of AdaBoost classifiers committee it is essential to use balanced sets of positive and negative samples, i.e. sets approximately equal sizes. However we have the set of negative samples (i.e. false regions) several times greater than the set of positive samples (i.e. red eye regions). A convenient way for reduction number of negative samples is applying of cascade of classifiers where first classifiers of cascade eliminate obvious false regions. The basic idea is to construct a cascade, which consists of some amount of trained classifiers (cascade layer). Each layer produces binary decision (accept/reject). If current layer rejects sample, then it is rejected by cascade in general and the classification procedure stops. If the sample is accepted, then it is passed to the latter layer of the cascade. Cascades are built hierarchically, each subsequent layer is more accurate then its predecessor. Figure 10 demonstrates our classification cascade where first 3 classifiers are applied to eliminate false regions.

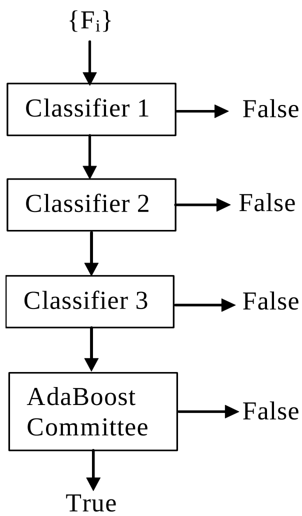


Figure 10: Classification cascade.

All first 3 classifiers are based on heuristic rules. We tried to apply machine learning techniques for building of these classifiers but outcomes of cascade in whole were worsen in comparison with heuristic rules.

Let's consider the first 3 classifiers of cascade. For every connected region a set of features is computed. Let (*r1,c1*) be coordinates of the left-top corner of the bounding box, (*r2,c2*) are coordinates of the right-bottom corner. Let *p(r,c)* be a function which is equal to 1, if pixel with coordinates (*r,c*) belongs to the connected region and is equal to 0 otherwise. The following features are calculated:

N is total number of pixels of connected region;

Nr is number of pixels having *color mark*;

N1, *N2*, *N3* and *N4* are numbers of pixels of a connected region marked with every of directional edge detecting filters;

Size of the bounding box:

$$L = c2 - c1, \tag{11}$$

$$W = r2 - r1; \tag{12}$$

Compactness:

$$Kc = \frac{N}{L \times W} \tag{13}$$

Elongation:

$$Ke1 = \frac{\min(L,W)}{\max(L,W)} \tag{14}$$

Number of directional edge detecting filters, whose indexes are stored in array of marks for a given connected region:

$$M = sign(N1) + sign(N2) + sign(N3) + sign(N4) \tag{15}$$

Ratio between maximal amount of pixels, marked with every of directional edge detecting filters, and total number of pixels in given connected region:

$$Ka = \frac{\max(N1, N2, N3, N4)}{N} \tag{16}$$

The first elementary classifier of cascade is the following condition: if *Nr* = 0 OR *M* < 2, then the region is rejected. This rule decreases number of FP significantly, after application of this classifier the average number of FP per image is 60. However part of positive red eyes are regions eliminated erroneously. Regions survived after first classifier of cascade are shown on Figure 11.

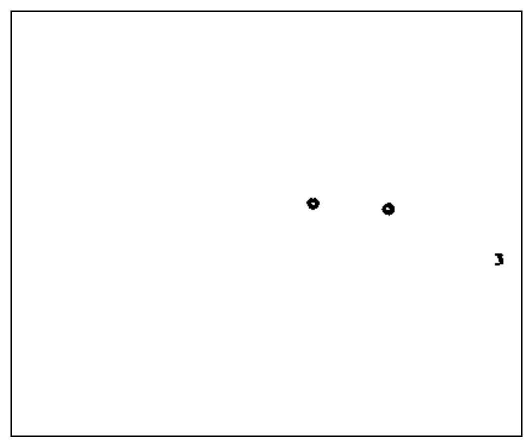


Figure 11: Regions survived after first classifier of cascade for photo from Figure 1.

Second classifier is decision tree with thresholds for features *M* *Ke1*, *Kc* and *Ka* > 0.5. After application of this classifier the average number of FP per image is 39. Number of FN increases insignificantly.

The following classifier employs a rule based on anthropomorphic ratio between the width of head and pupil of a human.

Initially we calculated for each surviving after previous classifiers of cascade region 80 features, which describe size and form of region, color of region and its vicinity, local contrast, percentage of skin tones in surroundings, percentage of tones non typical for human faces, similarity region to its vicinity, presence of whitish and highlighted pixels, percentage of edges in surroundings, relationships between sub-regions marked by directional filters an marked via 3D color tables, for each region survived after previous classifiers of cascade. Using the AdaBoost possibility to select relevant features we selected 48 features applicable for applying as weak learners. A computational complexity and memory requirement for calculation of different features differs significantly because we used only part of this feature set for building final classifiers for various implementations of proposed method. It is interesting that several features correlate with each other but reduction of correlated features a little bit worsens classification quality.

Also we experimented with different classifiers and its parameters. In general Gentle AdaBoost with CART with tree depth 3 provides better results. Figure 12 demonstrates Receiver Operator Characteristic (ROC) curve for one of the best version of classifier which was constructed containing 180 weak classifiers for 32 features, each weak classifier is CART with tree depth 3. The error rate is about 1%

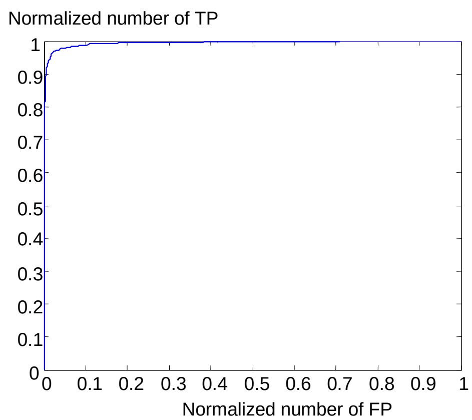


Figure 12: ROC curve for AdaBoost classifier.

5. DETECTION RESULTS

For estimation of detection quality, a quantitative quality criterion is proposed which was compared with existing automatic solutions. The retouching quality is important too but it is more subjective because we evaluated detection and correction quality separately.

Obviously that good solution have low FN and FP, ideally FN and FP are equal to zero. However severity of different FP differs significantly. Almost undistinguishable small FP on foreground is undesirable but sometimes allowable. Visible FP on foreground especially on human faces and bodies is unallowable absolutely; such FP artifacts damages photo more than red eyes. Therefore we divided all FP in two groups: FP_C are critical FP and FP_N are non-critical FP. FP_C decreases quality criterion more strongly than FP_N. Similar situation with FN; visibility and correction necessity of various types of red eyes are different. In our training and testing sets we labeled each red eye as HQ or LQ. Detection of HQ red eyes is essential whereas detection of LQ red eyes is desirable. We divided all FN in two groups: FN_{HQ} are FN of HQ red eyes and FN_{LQ} are FN of LQ red eyes. FN_{HQ} decreases quality criterion stronger than FN_{LQ}.

One more unwanted situation is correction of only one eye from pair. For software it is not so crucial because user has possibility to correct second eye manually, but for embedded implementation it is quite unpleasant for users.

Taking into account these points the following detection quality criterion Qc is proposed:

$$Qc = \frac{N_t - 2\times FN_{HQ} - FN_{LQ} - 5\times FP_C - FP_N - 2\times N_P}{N_t}, \qquad (17)$$

where N_t is total number of red eyes, N_P is number of faces with one corrected eye from pair of red eyes.

The following solutions with automatic red eye correction feature were tested: Nikon View 6.2.7 with Fotonation red eye detector; HP Photosmart 475 photoprinter, HP RedBot Web service (www.redbot.net); Kodak Easy Share 6.01; Canon Easy-PhotoPrint 3.4; Microsoft Digital Image Starter Edition 2006; ArcSoft Photoprinter 5.0 software utility; Cyberlink PhotoNow 1.0 (www.cyberlink.com) which uses OpenCV face detector.

For calculation of detection quality we used testing set containing 50 photos (N_t = 152, number of HQ red eyes is 104, number of LQ red eyes is 48). Table 3 contains comparison of detection quality of existing automatic red eye correction solutions and proposed methods. The table shows that proposed method provides the best outcomes in comparison with competitors because it has minimal FP and FN and maximal Qc for almost any weights in expression (17).

TABLE 3 COMPARISON OF DETECTION QUALITY OF EXISTING AUTOMATIC RED EYE CORRECTIONS SOLUTIONS

	FN_{HQ}	FN_{LQ}	FP_C	FP_N	N_P	Qc
Proposed	2	21	0	2	13	0.65
Fotonation	7	21	0	3	12	0.59
Arcsoft	10	22	0	2	13	0.54
HP RedBot	18	17	3	4	5	0.46
Canon	12	31	2	3	9	0.43
HP PS 475	11	27	4	9	14	0.36
Kodak	24	39	1	1	2	0.36
Microsoft	23	31	6	6	15	0.06
Cyberlink	30	27	21	18	6	0

6. CONCLUSION

Red eye artifacts are troublesome problem of consumer photos. Correction of red eyes during printing without user intervention is an important task.

The novel method of automatic detection of red eyes is proposed. This approach is based on application of color information via 3D tables and edge information via directional edge detection filters. For classification a cascade of classifiers including Gentle AdaBoost classifier is applied. Processing is performed in YCbCr color space but in general proposed method can be adopted for any color space easily.

The method is disclosed in two patent applications [30, 31].

7. ACKNOWLEDGMENT

Author would like to thank Dr. M. Rychagov, A. Shakenov and E. Tolstaya from SRC for discussions about algorithm and Dr. V. Vezhnevets and A. Vezhnevets from Graphics Media Lab of Moscow State University for GML AdaBoost Matlab Toolbox and advices. Thanks also go to Graphicon’s anonymous reviewers for their comments.

8. REFERENCES

[1] Brian W. Keelan. *Hadbook of image quality: Characterization and Prediction, Marcel Dekker, Inc., 2002.*

[2] Andrew Patti, Konstantinos Konstantinidies, Dan Tretter, Qian Lin. *Automatic Digital Redeye Reduction, International Conference on Image Processing, pp. 55-59, 1998.*

[3] S.Ioffe. *Red eye detection with machine learning, International Conference on Image Processing, vol. 2, pp. 871-874, 2003.*

[4] Peter Corcoran, Petronel Bigioi, Eran Steinberg, Alexei Pososin. *Automated In-Camera Detection of Flash-Eye, Defects IEEE Trans. on Consumer Electronics, vol. 51, pp. 11-17, Feb. 2005.*

[5] L.Zhang, Y.Sun, M.Li, H.Zhang. *Automated red-eye detection and correction in digital photographs, International Conference on Image Processing, ICIP '04, Vol. 4, pp. 2363 - 2366 Oct. 2004.*

[6] J.Y.Hardeberg. *Red Eye Removal using Digital Color Image Processing, PICS 2001: Image Processing, Image Quality, Image Capture Systems Conference, pp. 283-287, Monreal, Canada, 2001.*

[7] R.Schettini, F.Gasparini, F.Chazli. *A modular procedure for automatic redeye correction in digital photos, Proceedings of Electronic Imaging Science and Technology, IS6T/SPIE 16th International Symposium, USA, 2004.*

[8] F.Gasparini, R.Schettini. *Automatic Redeye removal for smart Enhancement of Photos of Unknown Origin, 8th Internetional conference on visual information systems, Netherlands, vol. 3736, pp. 226-233, 2005.*

[9] Wu. *Automatic red eye removal, US patent application 2005/0232481.*

[10] Luo et al. *Detecting and correcting redeye in an image, US patent application 2005/0047655.*

[11] K.Czubin, B.Smolka, M. Szczepanski, J.Y. Hardeberg, K.N. Plataniotis. *On the Redeye Effect Removal Algorithm, The First European Conference on Colour Graphics, Imaging and vision, France, pp. 292-297, 2002.*

[12] B.Smolka, K.Czubin, J.Y. Hardeberg, K.N. Plataniotis, M. Szczepanski, K.Wojciechowski. *Towards automatic redeye effect removal, Pattern Recognition Letters 24, pp. 1767-1785, 2003.*

[13] Huitao Luo, Jonathan Yen, Dan Tretter. *An Efficient Automatic Redeye Detection and Correction Algorithm, IEEE International conference on pattern recognition, vol.2, pp. 883-886, 2004.*

[14] Luo et al. *Detecting and correcting red-eye in a digital image, US patent application 2004/0213476.*

[15] A.Held. *Model-Based Correction of Red Eye Defects, 10th Color Imaging Conference: Color Science and engineering systems, Technologies, Applications, pp. 223-228, Arizona, 2002.*

[16] M.Gaubatz, R.Ulichney. *Automatic red-eye detection and correction, International Conference on Image Processing, vol. 1, pp. 804-807, 2002.*

[17] P.Viola, M.Jones. *Robust real-time object detection, Technical Report CRL 2001/01, Compaq Cambridge Research Laboratory, 2001.*

[18] Y.Freund, R. Schapire. *Experiments with a new boosting algorithm, International conference on Machine Learning, pp. 148-156, 1996.*

[19] R.E. Schapire, Y.Singer. *Improved boosting algorithms using confidence-rated predictions. Machine Learning, 37(3), pp. 297-336, 1999.*

[20] R.Ulichney, M.Gaubatz, J.M. van Thong. *RedBot - a tool for improving red-eye correction, HP Lab publication , 2003.*

[21] J.S.Schildkraut, R.T.Gray. *A Fully Automatic Redeye Detection and Correction Algorithm, International Conference on Image Processing, pp.801-803, 2002.*

[22] Chen et al. *Red-eye detection based on red region detection with eye confirmation, US patent 6895112, 2005.*

[23] Oberhardt et al. *Method for the automatic detection of red-eye defects in photographic image data, US patent application 20030044178.*

[24] R.Adams, L.Bischof. *Seeded Region Growing, IEEE transactions on Pattern analysis and Machine Intelligence, Vol. 16, No. 6, pp. 641-647, 1994.*

[25] Steinberg. *Method and apparatus for the automatic real-time detection and correction of red-eye defects in batches of digital images or in handheld appliances, US patent 6873743, 2005.*

[26] G.Gomez, E.Morales. *Automatic feature construction and a simple rule induction algorithm for skin detection, proc. of the ICML Workshop on Machine Learning in Computer Vision, pp.31-38, 2002.*

[27] W.Pratt. *"Digital image processing", John Wiley & Sons, Inc., 2001.*

[28] J.Friedman, T.Hastie, R.Tibshirani. *Additive logistic regression: A statistical view of boosting. The Annals of Statistics, 38(2), pp. 337-374, 2000.*

[29] A.Vezhnevets, V.Vezhnevets. *Modest AdaBoost – teaching AdaBoost to generalize better, Graphicon, pp.322-325, 2005.*

[30] I.V.Safonov. *Method of automatic red eye correction, Russian patent application 2006123847, 2006.*

[31] I.V.Safonov. *Method of automatic red eye correction, Russian patent application 2006132154, 2006.*

About the author

Ilia V. Safonov received his MS degree in automatic and electronic engineering from Moscow Engineering Physics Institute/University (MEPhI), Russia in 1994 and his PhD degree in computer science from MEPhI in 1997.

Since 1998 he is an assistant professor of faculty of Cybernetics of MEPhI while conducting researches in image segmentation, features extraction and pattern recognition problems.

Since 2004, Dr. I. Safonov has joint Image Enhancement Technology Group, Printing Technology Lab, Samsung Research Center, Moscow, Russia where he is engaged in photo, video and document image enhancement projects.