Evaluation of Red-eye Correction Tools

Application Note

July 2009

Table of Contents

Executive Summary 3
Introduction to Red-Eye Effects 3
Origin and Nature of the Red-Eye Effect 3
Red-eye Detection and Correction 5
Technology for Red-Eye Detection and Correction 5
Professional Photography and Retouching 5
Embedded Algorithms 6
Measuring Detection and Correction Performance 7
Performance Metrics 7
Global Algorithm Evaluation 11
Embedded Algorithm Evaluation 12
Side-by-side Product Comparison 12
Test Strategy Summary 13
Incorrect Methodologies for Assessing Red-Eye Algorithms Performance 13
Conclusion 15

Executive Summary

The red-eye effect is a common, yet always disappointing, result of photography that renders pictures unappealing and sometimes even useless. It also has the power to frustrate consumers and taint their perception of a camera or camera-enabled device.

This paper addresses the origin of red-eye effect and outlines best practices for testing red-eye removal features in any camera platform.

Luckily, with thorough understanding of the red-eye problem, application of the right software, and proper testing, cameras can be developed with high-quality red-eye detection and correction capabilities.

This paper examines the factors that play a role in the effectiveness of red-eye technology, such as subject variations and environmental conditions, the size and type of camera platform, and varying algorithms. When you know how to measure performance, set-up a definitive test and tune detection/correction algorithms, you will be able to efficiently incorporate the best red-eye correction capability into your products while boosting competitive differentiation and consumer satisfaction.

Introduction to Red-Eye Effects

Though red-eye detection and correction is a standard feature on most digital cameras today, the technology is not one-size-fits-all. Its effectiveness varies based on many factors, including lighting conditions, subject matter and camera angle. Understanding what causes red-eye effect is a good first step toward building a better camera.

Origin and Nature of the Red-Eye Effect

The red-eye effect is a negative outcome of flash photography, causing the pupils of the eye to appear red, or other bright color instead of black. The origin of the red-eye effect is well known and understood. The light from a flash source penetrates the pupil and illuminates the retina. The image of the retina is then projected back through the pupil onto the camera sensor.

Normally the pupil constricts when subjected to light. That is why a very small amount of light will reach the retina under normal lighting conditions. For this reason the eyes do not

appear red in natural conditions. With flash strobes, the red-eye effect appears because the pupil does not have the time to constrict so the retina is very well lit (Figure 1).

Figure 1 - *Under normal lighting conditions, the pupil is constricted so not enough light penetrates the eye to produce the red-eye effect. When the pupil is dilated and the flash fires, the pupil is not able to constrict for the short duration of the strobe. A large amount of light enters the pupil and lights the retina, producing the red-eye effect.*

However, the retinal region that is lit by flash must coincide with the one that is projected to the camera sensor in order for the red-eye effect to occur. This translates into the existence of a critical angle between the flash position, retina and camera lens, above which the effect does not occur (Figure 2). The value of this angle is approximately 3 degrees, but it varies with the subject's physiological characteristics as well as gaze angle.

Three parameters define this critical angle: the flash to lens distance of the camera, the camera to subject distance and the pupil dilation. Thus, the red-eye effect is most prevalent in photographs taken with compact digital still cameras and camera-enabled cell phones where the flash is normally very close to the lens. Additionally, the incidence of red eye increases with subject distance and pupil size. The latter relates to the amount of light in the field of view of the subject. In daylight, when the pupil is almost as small as a pinhead the red-eye effect appears very scarcely. Conversely, the pupil is dilated in dark environments so the chance of the effect to occur is high.

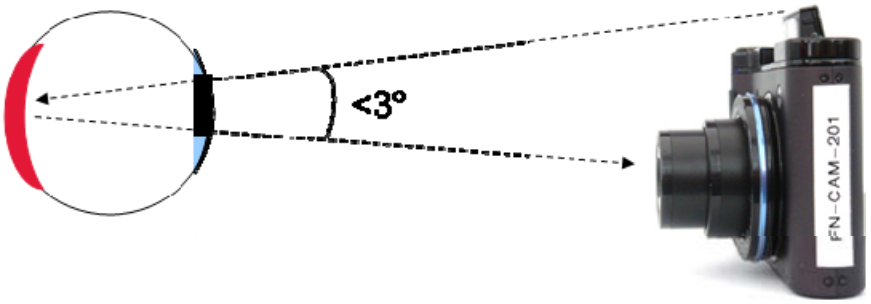

Figure 2 - *The red-eye effect occurs when light from the flash source is reflected off the retina back into the camera, which requires a small angle between the light vectors.*

The red-eye effect does not always result in pupils being colored red. Under certain conditions, the pupils appear orange or gold, a phenomenon commonly called "golden eye", or half red and half orange. These types of defect have a higher incidence in people of Asian

origin. There are also other types of flash defects, such as white or purple and they sometimes have a non-uniform appearance. In this paper, the term “red eye” will be used to describe all unnatural pupil coloration resulting from the red-eye effect.

***Figure 3** - Different manifestations of the red-eye effect: from left to right red, golden, half-red, other*

Red-eye Detection and Correction

Red eye is ubiquitous in flash photography. Photographs that exhibit some form of red eye are visually displeasing and usually discarded. Therefore, the ability to eliminate this unwanted effect is a highly desirable feature for cameras.

The presence of the red-eye effect in digital pictures can be detected and corrected with software. Furthermore, this type of software can be embedded on mobile platforms, allowing red-eye prevention features to be built directly into cameras. Red-eye correction can be completely automatic, adjusting pictures as they are taken. However, the effectiveness of algorithms to address the red-eye effect is subject to statistical variation. In some cases they fail to detect the red-eye effect, make a visually poor correction, or alter a region of the photograph that is not an eye, such as circles or dots on clothing. A rigorous methodology, underpinned by an understanding of the origin of the red-eye effect, is required to correctly evaluate the performance of red-eye removal tools.

Technology for Red-Eye Detection and Correction

Camera devices contain many different features and specifications, based on brand, model, function and size. As an example, the flash can be located in various positions with respect to the camera lens.

Understanding and applying the appropriate red-eye detection and correction technology to specific products will greatly improve the performance of camera as well as the overall customer satisfaction in the product.

Professional Photography and Retouching

Professional photographers are generally aware of the conditions that contribute to the red-eye problem. This is why more expensive cameras used in studio photography have a flash gun mounted on an arm or remote stand, so the photographer can eliminate any chance of red-eye effects. The purpose of separating the equipment is to increase the angle between the flash source and the optical axis of the camera lens so that it always exceeds the red-eye critical angle. Another common technique involves pointing the flash source upward so the light bounces off of the ceiling before reaching the subject.

Some cameras use a multiple flash sequence to eliminate the red-eye effect. In these cameras, the photograph is captured on the last flash. The basis of the approach is to force the pupils to contract, which decreases the risk of red eye. Also, the auto-focus assist lamp is sometimes used to reduce the incidence of the red-eye effect. As this lamp is lit for relatively long periods of time (usually seconds), the pupils have enough time to contract before the photograph is taken. However, the efficiency of this method is very low when the subject does not look toward the camera.

Unfortunately on many compact digital cameras and cell phones, ample physical separation between the flash gun and lens is not possible. Additionally, as the efficiency of traditional red-eye reduction techniques such as pre-flashes and red-eye reduction lamps is limited, the red-eye effect can not be altogether eliminated.

Computer programs that manipulate digital photographs, including red-eye removal, have been available for many years. However, their use requires purchase of software, a computer with a good specification, and the time and expertise to operate the software and transfer the photographs from the mobile platform to the computer. While this is acceptable for professional photographers, most casual photographers want better quality images automatically. The ideal approach for camera manufacturers is to embed fully automatic software on the mobile platform that seamlessly corrects affected photographs without any user intervention.

Embedded Algorithms

A major variable that affects the performance of red-eye detection and correction algorithms is the device on which they are embedded. For example, generic red-eye software which is effective on a vast database of reference photographs may not necessarily be optimal for a specific camera model.

Moreover, in some PC applications the user can interact and help the algorithm (e.g. by undoing the failures or increasing detection strength). In most of the in-camera use cases, however, this is not possible.

Every camera device has nuances that may affect the behavior of the algorithm. This variation can be demonstrated by capturing the same scene using a range of cameras. Even if the images are the same resolution (pixel count), appreciable differences will exist in such attributes as contrast, brightness, color gamut, distortion and noise, with additional variation caused by differences in the flash sources (color spectrum, intensity, angle to camera axis, etc). Due to this inherent variability, red-eye algorithms might require tuning for certain camera models.

In addition to the algorithms, there are many high-level factors to consider when selecting the appropriate red-eye removal technology to use in specific cameras, including:

- Code size (ROM)
- Memory requirement for the code to operate (RAM)
- Execution speed (factoring clock speed and available MIPS)
- Dependency on the platform operating system
- Application programming interface requirements
- Picture resolution
- Minimum/maximum number of eyes and eye size per picture
- Availability of "undo" option
- Detection rate
- False positive rate

Choosing technology with suitable specifications for individual camera platforms is critical to the effectiveness of the feature.

Measuring Detection and Correction Performance

Because many factors affect how a specific camera functions—and, specifically, how red-eye technology operates on different devices—it is imperative to establish an accurate process for measuring red-eye detection and correction performance in every product. Carefully measuring red-eye performance in different situations will help ensure quality control on camera products.

Performance Metrics

Following are three possible outcomes from use of a red-eye removal algorithm applied to a photograph:

1) Red-eye detection and successful correction (Figure 4)

Figure 4 - *Successful red-eye detection and correction*

2) No red-eye detection and correction applied. Possible causes include:

- There are no red-eyes in the photograph (Figure 5)
- The red-eye algorithm fails. None of the current red-eye solutions achieve a 100% detection rate, so behavior is not unexpected. The red eyes which are not detected by the algorithm will be denoted as ***False Negatives*** or ***FNs***. (Figure 6). The failure rate of a red eye algorithm depends on internal and external factors:
  - Algorithm limitations (distance to subject range, unusual face angles for algorithms which are based on face tracking, algorithm overall performance itself)
  - Poor image quality (bad white balance, color matching, too much noise, bad exposure)
- The red-eye solution malfunctions due to system integration errors (e.g improper usage of the library API)

Figure 5 - *The red-eye effect is not present in the photograph*

Figure 6 - *The red-eye effect is present in the photograph, but the algorithm fails to correct it*

3) Red-eye misdetection or ineffective correction. Under certain circumstances, all red-eye correction algorithms will exhibit such behavior. However this must happen very scarcely and the result should not be too striking. These misdetections will be further denoted as ***False Positives*** or ***FPs***.

Figure 7 - *The region around the eye is incorrectly darkened instead of the red pupil*

Figure 8 - *The red cap of the pot is incorrectly classified as red eye and rendered black*

Figure 9 - *The red lip is mistaken for a red eye and it is processed accordingly*

In order to evaluate the performance of a red-eye correction solution, the following metrics are commonly used:

Algorithm Success Rate = # corrected eyes / # defect eyes x *100*
Overall Camera Success Rate = (# eyes - #FNs) / #eyes x *100*
False Positive Rate = #FPs / #Photos x *100*

where:

- *#eyes* – The total number of eyes in the evaluated photo set
- *#defect eyes* – The total number of eyes affected by the red eye effect.
- *#corrected eyes* – The number of red eyes corrected successfully by the algorithm
- *#FNs* – The number of eyes that have been affected by red-eye but have not been corrected.
- *#FPs* – The number of misdetections.

The first metric is an indication of the performance of the red-eye algorithm on the entire set of images. For example, let us consider a set of 300 photos with 800 eyes out of which only 50 eyes are defect. If the algorithm manages to correct a total of 10 defect eyes, then the success rate of the algorithm would be 20%. However this is not an adequate indication in all cases of what the customer actually gets.

Thus, the second metric was defined to give a comprehensive indication by describing the performance of the overall camera-algorithm duo. On the entire set of 300 photos, there are 40 red eyes out of 800. This yields an overall camera performance of (800-40)/800 = 95%.

Additionally, the quality of the correction can be assessed by visual inspection. Usually, a human expert compares the corrected image against the original and assigns grades based on the esthetics of the correction. The expectations and consequently the rigorousness of the grading should increase with the quality of the input image.

Figure 10 - *Correction quality evaluation aims at quantifying the life-like appearance of the corrected eyes. In the figure, the second correction has a light, unnatural tone compared to the first.*

Here is an example of a grade scale that is normally used for assessing the correction quality:

- 3 – the correction appears natural and is pleasing
- 2 – the correction appears unnatural, but it is better than the original red eye
- 1 – the correction appears worse than the original red eye
- N/A – no red-eye correction has been performed

The above scale is normally applied on each eye, individually. Then, in order to determine overall grades (image-wise or on entire database), the grades are averaged.

Global Algorithm Evaluation

A global evaluation of a red-eye correction software aims at determining its generic performance rather than in relation to a certain camera platform. The key to doing that is testing on a very large database of images covering a vast diversity of cameras, subjects and shooting conditions, as well as scenes engineered to trigger false positive responses. Usually, such a database should encompass tens of thousands of images to be statistically relevant.

However the mere number of images is not enough to guarantee that an accurate conclusion about the performance of the algorithm can be drawn. One must keep in mind that the red-eye effect appears in an infinite number of situations and that there are many factors that influence it. Consequently, a good database should cover as many of these situations as possible.

Some of the parameters that should be accounted for when building such database are:

1. Camera dependent parameters – flash position, lens characteristics, sensor characteristics, image processing
2. Subject's dependent parameters – ethnicity, age, sex, eye conditions (such as eye health or the presence of corrective optics)
3. Photo shoot dependent parameters – subject distance, framing, subject pose, gaze angle, background, lighting

Once a valid database has been prepared, the testing methodology consists of applying the red-eye correction algorithm on the photos, analyzing the results and calculating the detection rate and false positive rate. However, in order to do this automatically, the images in the database should include metadata which indicates the positions of the red eyes along with the nature of the defect. This information is usually called *image markings*. An automated tool that processes the images by applying red-eye correction should identify the changes performed by the algorithm and then compare these changes against the image markings. A change that overlaps with an eye can be said to be a properly identified red eye, while a change that does not would be a false positive. The downside of this approach is that the correction quality is disregarded.

Unfortunately, correction quality is very difficult to assess on such large databases, since this involves human subjective evaluation. For this purpose a smaller set, usually around 2000-5000 photos, with a good distribution can be used. Only photos for which the detection succeeds are normally included in this database. This is because the quality of correction can only be evaluated on eyes that have been properly detected.

Embedded Algorithm Evaluation

A global evaluation aims at determining the general performance of a red-eye correction solution. However, such evaluation does not indicate how the algorithm performs in a certain embedded device. For example, a solution that does not score too well on a large database with tens of thousands of photos might give very good results for a certain camera model. This is because in some cases, embedded algorithms are specifically calibrated for some devices with uncommon features in terms of image quality.

An embedded evaluation has the role of determining how well a red-eye correction algorithm works in a certain camera-enabled device. The following methodology is required for a typical embedded evaluation:

1. Make sure that you use the production level device. You will not get relevant results if you use intermediate prototype versions or any device with noticeable image quality issues.

2. Create a set of photos with the actual hardware device. For that, a photo shoot with approximately 10 subjects of various ages, sex, eye color and ethnicity is required. The photo shoot must cover as much of the camera range as possible. Depending on this range, the total number of photos in the database will be approximately 300-500.

3. Apply the correction in-camera and save both original and corrected images. It is very important that the two images are saved identically, because no other differences except the ones performed by the algorithm should exist when performing the comparison.

4. Analyze the photos and compute the necessary statistics. Among all metrics, one is of extreme interest in this case: the *Overall Camera Success Rate*. This shows how many red eyes the user actually sees.

Side-by-side Product Comparison

Sometimes a side-by-side comparison of two different red-eye removal algorithms is desired. The common misunderstanding is that if two images are taken in the exact conditions, then they must be identical; consequently they can be used in a side-by-side product comparison.

Given the nature of the red-eye effect and its appearance in photography, this assumption is fundamentally incorrect. First of all, the slightest difference between two imaging devices such as a different noise reduction level can have a profound impact on the aspect of the red eye. Moreover, even if the same device is used to take two images in the same conditions, the two images cannot be identical. A slight change in focus, a change in the pupil size or eye gaze can make a huge difference. Sometimes, even the jpeg encoder/decoder used for processing the photos can introduce a significant amount of variability. Consequently, side-by-side product comparisons tend to have questionable value if not done properly.

To do an accurate side-by-side red-eye embedded evaluation, use the target platform to create a single image collection for all algorithms. Do not enable any red-eye correction algorithm in this phase and save the uncompressed images. This is a compulsory requirement, as the raw image data is normally the input of the algorithm. Each of the solutions under evaluation must

then be integrated in the same platform and applied on the image set. The output should then be saved and analyzed. You will now have a reliable comparative analysis of the different technologies.

Test Strategy Summary

As discussed in the previous section, numerous factors and conditions need to be considered and evaluated when testing red-eye removal technology. The following is a summary of best practice test strategy:

Test Type	Test Purpose	Test Procedure
Global Evaluation	Assess how well a red-eye correction algorithm functions using a large and categorized database of images	1. Obtain large database (>10,000 entries) of categorized images. 2. Run red-eye removal tool on each image and grade quality of result. 3. Statistically analyze findings.
Embedded Validation	Quantify how well an algorithm works with live scenes when embedded on a mobile platform	1. Port and tune algorithm to a production-ready platform. 2. Capture a set of 300 images or more, varying as many of the parameters that influence the red-eye effect. 3. Compute success and false positive rates.
Product Comparison	Compare the performance of one red-eye technology against another	1. Do not use different platforms or different image sets as the result is subject to gross error. 2. Capture 300 images or more with the target platform (no red-eye correction enabled) and save raw images. 3. Integrate each of the technologies in the target platform and apply correction. 4. Statistically analyze findings and compare results.

Incorrect Methodologies for Assessing Red-Eye Algorithm Performance

Capturing the required number of live images with the requisite diversity presents a major logistical challenge. Some manufacturers attempt to validate a red-eye removal algorithm using inanimate subjects, such as mannequins, paintings of faces or computer-generated faces, to build their portfolio of images. All these techniques are not valid for evaluating the performance of a red-eye algorithm.

Taking photos of prints or displayed images is one of the most common mistakes in red-eye algorithm evaluation. Most red-eye algorithms are intelligent enough to distinguish a real human eye from an artificial image. The following example shows the differences between a

real image, an image photographed from a print and an image photographed on an LCD display:

Figure 11 - *Differences between a real photograph (left) and photographs taken on print (middle) and LCD display (right). Red-eye algorithms are affected by such differences, consequently using prints and displayed images is an incorrect testing technique.*

The difference between the three is striking even to the human eye. Details in the original image are lost, artifacts are introduced, contrast is altered, color casts appear, etc. Under these circumstances it is no wonder that most red-eye algorithms fail in identifying such images as being real red eyes.

Another common bad practice is to engineer red-eye images by painting red circles over non-red eyes. Although these artificial defects might obey the definition of a red eye, they cannot replicate the fine details of a human red eye, as Figure 12 shows.

Figure 12 - *Fake images generated by painting red circles over non-red eyes cannot reproduce the fine details of real red eyes.*

It is possible to make special mannequins that will generate a true red-eye response (Figure 13). The mannequin's eyes appear red (or another color such as golden) when the illumination conditions are suitable for the red-eye effect. While mannequins deliver a consistent red-eye response, they do not mimic the fine details of the human anatomy which are of paramount importance for red-eye removal algorithms. Therefore they cannot be used to conduct a proper evaluation of any red-eye removal tool. Mannequins are best saved for product demonstrations and to check other performance parameters such as execution time or memory requirements.

Figure 13 - *Mannequin engineered to generate the red-eye effect with flash photography. Typically, mannequins are used for demonstration and quality control purposes. Left: red-eye effect. Right: after application of the FotoNation Red detection and correction solution.*

Conclusion

As illustrated in this application note, red-eye removal technology is a necessary component for any mobile platform, but not all red-eye technology is created equal or works the same in different devices. The development of a camera platform with red-eye prevention capability involves understanding the red-eye effect and considering test conditions, functions to test and performance metrics to measure.

Selecting the most appropriate red-eye removal product to incorporate in your products is an important decision that requires careful evaluation.

With thorough understanding of the issue, application of the right software and proper testing, camera devices can be developed with high-quality red-eye detection and correction capabilities. As a result, manufacturers can satisfy consumers with better products produced more efficiently.

Tessera FotoNation® Red

Licensed for use in more than 100 million cameras and imaging products, Tessera's FotoNation Red red-eye detection and correction solution provides the highest red- and golden-eye detection rate and lowest false positive rate available today.

Contact a Tessera sales representative for more information or to schedule a FotoNation Red demonstration.

TESSERA

3025 Orchard Parkway • San Joes, CA
Tel: 1+408.894.0700 • Fax: 1+408.894.0768
www.tessera.com

