

FUJIFILM X10 white orbs and DeOrbIt

Henry Gordon Dietz*

University of Kentucky, Lexington, Kentucky, USA

Abstract. *The FUJIFILM X10 is a high-end enthusiast compact digital camera using an unusual sensor design. Unfortunately, upon its Fall 2011 release, the camera quickly became infamous for the uniquely disturbing "white orbs" that often appeared in areas where the sensor was saturated. FUJIFILM's first attempt at a fix was firmware released on February 25, 2012 – it had little effect. In April 2012, a sensor replacement essentially solved the problem.*

This paper explores the "white orb" phenomenon in detail. After FUJIFILM's attempt at a firmware fix failed, the author decided to create a post-processing tool that automatically could repair existing images. DeOrbIt was released as a free tool on March 7, 2012. To better understand the problem and how to fix it, the WWW form version of the tool logs images, processing parameters, and evaluations by users. The current paper describes the technical problem, the novel computational photography methods used by DeOrbit to repair affected images, and the public perceptions revealed by this experiment.



Figure 1. FUJIFILM X10, a "White Orb," and repair implemented by DeOrbIt

1. INTRODUCTION

The appearance of "white orbs" in photographs is not a new phenomenon. Various WWW sites claim they are images of ghosts or angels, while others say that it is just light reflecting off airborne dust. Some white orbs are merely the point spread function (PSF) of the lens created by out-of-focus (OOF) point light sources. Unfortunately, the FUJIFILM X10 has become infamous for yet a different type of white orbs, which this paper will simply refer to as "WOs" (pronounced "woes" in oral presentations). These WOs are apparently the result of sensor bloom.

Many sensors suffer charge leakage between adjacent pixels, but the X10's sensor has an unusual design which apparently allows exceptionally large blooms to occur under certain circumstances. These orbs are thus roughly circular regions of saturated pixels, often outlined by dark pixels thanks to overly aggressive sharpening. Figure 1 shows the FUJIFILM X10, a tight crop of an actual WO, and the same WO as repaired by DeOrbIt.

The rest of Section 1 provides an overview of the X10 camera, the unusual sensor it employs, and what other cameras suffer the same WOs. A detailed discussion of the problem appears in Section 2, including characterization of the phenomena and discussion of FUJIFILM's attempts to fix the problem. Both manual and automatic methods for repair of WOs are discussed in Section 3, along with a walk-through of the major functions in the C implementation of the DeOrbIt tool. Things learned from tool users, as well as samples of the WO repair results, are given in Section 4. Brief final comments are made in Section 5.

* Professor in the University of Kentucky's Department of Electrical and Computer Engineering, but also a member of the Center for Visualization and Virtual Environments. E-mail: hankd@engr.uky.edu

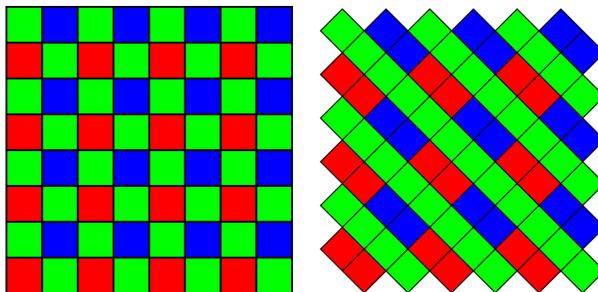


Figure 2. Bayer and FUJIFILM X10 CFA patterns

1.1 Overview Of The X10

With so many digital cameras in production, why is there so much interest in the FUJIFILM X10? An obvious reason is that it is a retro-styled entry on the leading edge of a fast-growing field of enthusiast-targeted compact cameras. The manual-zoom lens is yet another way this camera seems special; the lens covers the equivalent field of view of a 28-112mm lens on a full-frame 35mm camera, and does so with a fast $f/2.0$ - $f/2.8$ maximum aperture. Although the X10's optical finder lacks a rangefinder, it would be easy to mistake the X10 for a high-end rangefinder film camera made 25 years ago; both the styling of its all-metal body and the proliferation of manual controls are clearly intended to reinforce the idea that this competes more with film cameras from Leica than with digitals from Nikon, Canon, and Sony. This notion is furthered by features like well-implemented "film simulation" modes that yield JPEG images precisely mimicking the subtle differences between, for example, Velvia and Astia films.

From a more technical point of view, the basic specifications of the X10's sensor and digital processing at first appear rather ordinary. The 12MP CMOS $2/3"$ ($8.8 \times 6.6 \mu\text{m}$) sensor does not sound particularly innovative; there are higher resolution and/or larger sensors in competing models. However, the X10's sensor has an unusual design.

1.2 The X10 Sensor

Like various other FUJIFILM cameras, the sensor in the X10 uses a diagonal layout of pixel sites rather than a much more common rectangular pattern. It is standard engineering practice that the ability to distinguish colors is implemented by a color filter array (CFA) placed in front of the sensor, and the X10 also uses this approach, but with an innovative pattern. A conventional Bayer CFA looks something like the left image in Figure 2. Instead, the X10's "EXR" sensor uses a CFA pattern in which each pixel of any given color always has at least one neighboring pixel of the same color, as shown in the right image of Figure 2. FUJIFILM does not make clear how much the architecture of the sensor chip itself is tuned to take advantage of this CFA pattern, but the camera as a system supports processing the sensor data in three distinct ways:

HR Mode Full color information is interpolated for each pixel site, providing high resolution – the full 12 million pixels.

DR Mode Pairs of adjacent same-color pixels are treated as single, high dynamic range, pixels. Within each pair, one pixel is exposed normally while the other is sampled early, thus hopefully avoiding saturation. The result is an image containing 6 million pixels.

SN Mode Pairs of adjacent same-color pixels are averaged to produce lower-noise pixel values. The details of the averaging process are not given; it is not even clear if averaging is done in the digital or analog domain. The result is an image containing 6 million pixels.

These modes do seem to allow the X10 to outperform competitors using otherwise-comparable more conventional sensors.

Unfortunately, the WOs problem which is the topic of this article is almost certainly caused by the sensor and/or the interpolation process required to leverage the unusual CFA. The firmware must implement an interpolation process which is less well understood, and less symmetric, than with a conventional rectangular Bayer pattern, and this could result in difficult cases being handled less well. However, the DR and SN modes both suggest somewhat unusual sensor wiring and/or circuitry may be present, and it is easy to imagine that this may be the root cause of the WOs. There are several pieces of evidence supporting this speculation:

- In a DPReview article,² it was reported that there was virtually no difference in applying a firmware update intended to fix the WOs problem to any particular X10, but there was a dramatic difference in the severity of the orbs across different *samples* of the X10. This suggests the problem lies in properties of the sensor which are affected by semiconductor fabrication variations – such as analog properties of the sensels or analog to digital converters.
- When FUJIFILM claimed to have a definitive fix for the WOs problem,⁴ the fix involved sending your camera to a service center for “sensor replacement.” This might not have been necessary if the problem was entirely in the firmware processing sensor data. Further, FUJIFILM’s sales packaging of the X10 does not distinguish between the two sensor versions and the latest firmware update can be installed on cameras having either version.
- It is well known that the WOs are smaller at higher ISO settings, especially above ISO 400. This suggests that the analog charge storage per sensel is insufficient to allow even modest overexposure at lower ISOs. Of course, photon shot noise and higher amplification reduce signal/noise ratio at higher ISOs, so there is strong motivation for the camera to allow lower ISOs.
- The WOs problem is present in raw, uninterpolated, images – not just JPEGs.

Although it seems that the new sensor eliminates WOs per se, it still appears to have a somewhat different handling of saturated regions than most cameras.

1.3 Cameras With Similar Issues

Although the WOs problem is generally identified with the X10, FUJIFILM uses the same sensor in the X-S1 and it suffers the same issues. Perhaps the fact that the X-S1 is a consumer-targeted superzoom, rather than a compact targeted at “prosumer” users, explains why the problem is less tolerated in the X10?

Are there other cameras in other brands, using different sensors, that suffer the same WOs problem? Despite searching reviews of hundreds of cameras, and having personally evaluated dozens of different types of digital cameras, I was unable to find or create a single instance of a fully-formed WO in images captured by any other camera.

This is not to say that the X10 and X-S1 have the worst saturation artifacts – far from it. For example, many older cameras suffer horizontal or vertical saturation stripes resulting from charge leaking from saturated sensels along the entire path to the analog-to-digital converter. Unpleasant saturation artifacts involving corruption of color information are even more common in older cameras, generally resulting from applying the usual interpolation formulas when one or more channels, but not all, have saturated.

Given the above, it may seem strange that the current paper claims to repair the WOs problem using only software postprocessing. In fact, this problem cannot be repaired in the sense that the scene data lost by WOs cannot be recovered. Fortunately, credible repair does not require accurate recovery of lost data.

2. THE PROBLEM IN DETAIL

In order to design the best possible repair, it is useful to more precisely characterize how WOs appear in images. The `DeOrbIt` tool was initially developed without access to an X10, so examples had to come from other sources. Fortunately, hundreds of sample photos were made freely accessible as unhappy X10 owners and photography review WWW sites posted images demonstrating the extent of the problem. Images posted by DPReview² were particularly helpful in that they effectively isolated the WOs from other potentially confusing phenomena. As



Figure 3. “Black Hole” examples; Left: full image, Right: crop showing OOF PSF and hole detail

the first version of DeOrbIt was posted, it was made available as a CGI script accessed through a WWW form and users were told that images would be collected along with the user’s scoring of how well the tool corrected each image. Images could be submitted anonymously or attached to an email address; users were promised that images would not be published without their explicit permission via email (which many gave).

The insights gained through these hundreds of images were sufficient to create and tune the tool, but were not really sufficient to cleanly demonstrate the characteristics with just a few images. Thus, for this paper and live demonstrations at the conference, it was decided that purchasing an X10 with the original sensor and firmware would be appropriate. This was a more difficult task than one might expect, because there is no way to tell if a new camera purchased after the new sensor became available has the old or the new sensor. A little research showed that in North America, serial numbers starting with 14A were generally the original sensor; the used X10 obtained via eBay has serial number 14A00897 – a very low number. The firmware installed is version 1.02, which predates any attempts to fix the unfortunate saturation behaviors of the camera. This is the camera used for the images not otherwise marked in the current paper.

2.1 Black Holes?

Immediately upon arrival of our X10, a series of informal test shots were taken to confirm that this particular X10 did indeed have the WO problem. It did. However, it also has a much more severe “Black Hole” (BH) problem, which surfaced in the image of Figure 3; it is most obvious in the right side of the enlarged crop.

Like WOs, BHs result from bad handling of saturated areas. The visually obvious difference is that in a BH, the center of a saturated region is made solid black, and it turns to lighter grays, and finally white, as it approaches the edges. Unlike WOs, BHs only require saturation – they can occur in saturated regions that have not bloomed. Our testing found BHs in Velvia film simulation mode, but with contrast turned-up, BHs were even more common than WOs. It seems most likely that the high-contrast setting on top of the naturally high contrast of Velvia film caused a wrap-around rounding error in which a maximally bright pixel became black. The grays then most likely result from the interpolation process attempting a smooth transition from black to white.

Not surprisingly, the BH problem also was known on the Internet, and very early on had been a source of much concern. According to FUJIFILM’s “X10 Firmware Update” page,⁵ the BH problem was fixed with the release of the Ver. 1.03 firmware:

The image area with high brightness might be recorded as "black-out" area, just in case of dedicated shooting condition, This unpleasant phenomenon is improved.

For purposes of the current research, it was decided not to upgrade from the Ver. 1.02 firmware this X10 holds.



Figure 4. Actual WO for an in-focus point source, increasing exposure in approximately 1-stop increments

2.2 Characteristics Of WOs

One of the biggest issues with WOs is how often they occur: some X10 users claim never to have seen one, while others find them spoiling a large number of shots (there is more about user perceptions in Section 4). In test shooting with the above mentioned X10, approximately 20% of all images captured had at least one identifiable WO somewhere in the frame. About half of those, or 10% of all images captured, had WOs that were sufficiently visually distracting so as to warrant removal in post-processing. Although the BH phenomenon occurred in nearly 100% of images shot in Velvia film simulation mode with high contrast set, it never happened when that combination of settings was avoided.

Given the opportunity to perform specific tests using an X10, it also was possible to better define how WOs appear. Perhaps the most useful insight is that WOs truly are roughly circular, independent of the position within the frame, ISO used, etc. Other shapes occur only by overlap between WOs and other saturated areas. They also are essentially white, not appearing in any other color despite significant changes to the color of the saturating light source. Even using a pure red or blue LED light source generates a WO, although lens flare is colored as the light source. The saturated pixel values within a WO are not always the maximum representable value, but are within a few percent. Perhaps they differ from pure white due to dark noise subtraction or even simple fabrication differences between sensels?

The diameter of a WO appears to be a fairly straightforward function of how overexposed the seed point in the scene is. To study this relationship, a sequence of *in-focus* images were taken of a simulated point light source (white LED at 10m distance) in a darkened room. Crops from images spanning a roughly 21-stop EV range are shown in Figure 4. This sequence is stunning in that the point is only about 3 pixels wide, but blooms to a diameter of 86 pixels. That is spreading saturation over $800\times$ the original area!

Still more can be learned by careful examination of the edge of a WO – for example, the 86-pixel diameter one in Figure 4. First, there are traces of color around the WO. As they do in this example, the color fringing from a white light source commonly favors shades of green and blue. Second, there is a subtle dark outline, approximately 2 pixels wide, surrounding the WO. This dark outline is most likely the result of sharpening. If either of these effects is larger than expected and parameters are not adjusted, these minor defects can be carried into larger areas by DeOrbIt’s processing, thus making them become much more obvious defects.

2.3 FUJIFILM’s Fixes

Although the work reported in this paper is clearly relevant for X10 cameras using the original sensor with firmware Ver. 1.02, has FUJIFILM since made the processing reported here unnecessary?

A later announcement from FUJIFILM⁴ added that Ver. 1.03 attempted to fix the WOs:

In February, we announced a firmware upgrade for the X10 (version 1.03) which does reduce the white disc occurrence specifically in EXR mode. It works by identifying scenes that are likely to get ‘white disc’ blooming. When the camera recognizes such a scene, it automatically increases the ISO and optimizes DR (dynamic range). As the ISO increases, the white discs are less evident.

However, as reported by DPReview,² this change made virtually no difference. In fact, it probably failed to increase the ISO in most cases tested, literally changing nothing. There is no mention of further fixes in Ver. 2.00 of the firmware.

The new sensor which FUJIFILM is now supplying⁴ is said by nearly all reviewers to eliminate the WO problem. Without having an X10 with the new sensor to test, no images were found on the WWW clearly showing a WO from a camera after sensor replacement. The new sensor may still be more prone to bloom than is typical, but the distinctively disturbing WOs appear to be gone.

Of course, it is useful to note the obvious fact that replacing the sensor does nothing to remove WOs from photos already taken....

3. REPAIR ALGORITHM

The initial question was: would it be possible to repair the WOs? Clearly, true repair is a theoretical impossibility. There is no way to recover the information lost in saturated pixels. The goal is thus *credible* repair. This was first attempted by manually applying existing algorithms and then by coding a fully custom algorithm.

3.1 Manual Repair (gimp)

Generation of credible image content for a missing portion of an image is a well-studied problem. The most commonly used method for texture synthesis involves “cloning” nearby regions. However, it is very common that bright lights and specular highlights – the prime causes of WOs – occur at points near transitions between different textures. Simple cloning does not offer a mechanism by which a credible transition between textures can be synthesized.

Harrison’s Ph.D. dissertation⁶ provides not only a good overview of the texture synthesis problem, but also a very effective algorithm that was implemented as the “Resynthesizer” plug-in for `gimp`. As it is used in the “Heal selection” plug-in,⁷ this algorithm synthesizes random, yet highly credible, replacements for any selected region of an image. The algorithm seeds the region with values from nearby pixels and then fills-in between them by picking pixel values that appeared in a similar context nearby. This not only generates credible textures, but also transitions between them.

An initial test using `gimp` was performed by:

1. Select the regions of the original image obscured by WOs. As a human, this was a simple matter of using a color-region selection tool to click on each WO, followed by minor enlargement/cropping of the selected region to fully contain all the WOs and nothing else.
2. Synthesize credible background image content for the selected region using the “Heal selection” plug-in. Note that the synthesis process will tend to continue artifacts from outside the WOs into them – for example, a pointed flare pattern surrounding a WO will be drawn into the synthesized background replacing the WO.

After applying the above steps to one of the DPReview test images, I posted links to original and processed crops at <http://aggregate.org/DIT/Fuji0rb/> in a comment on the DPReview article describing FUJIFILM’s ineffective firmware fix.² My comment ended with: “Is this good enough to be worthwhile?” The community answered yes, but I was not quite convinced.

Although lens flare patterns from outside the WO being carried into the WO area often lead to synthesis of a central bright spot, they do not generate any pixels brighter than ones outside the WO. Clearly, at the center of each non-trivial WO should be a saturated pixel. Further, the entire area of the texture synthesized to replace

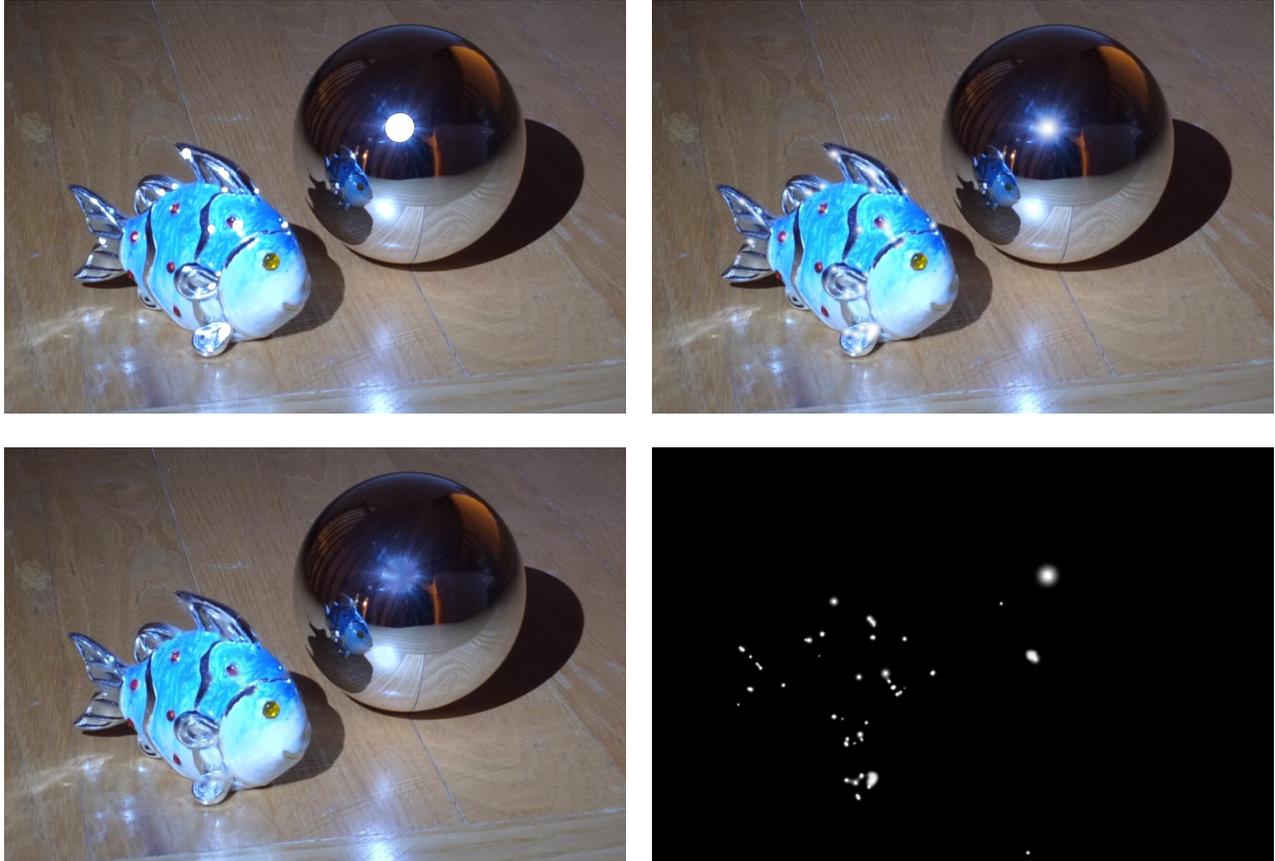


Figure 5. A Fishy Image: Original, Repaired, Synthesized Texture, Lighting

each WO is left fully visible. When textures are not credible, this is offensive. When texture synthesis has been very successful, there is the even more disturbing fact that significant details in the image can be entirely false without any obvious indication to the viewer. Thus, it became clear that a “relighting” step should be added to brighten the center and partially obscure the synthesized texture.

3.2 Automatic Repair (DeOrbIt)

The `DeOrbIt` tool implements both texture synthesis and relighting, but with another big difference: it is designed to be *fully automatic and computationally cheap enough to be implemented within the X10 camera itself*. There was also the issue that people might not agree on which of various alternatives was the best repair, so it was decided that `DeOrbIt` should initially be made available in a form that collected test images and allowed users to score quality of results.

The remainder of this section details how `DeOrbIt` works. There are three major steps discussed in the following three subsections: identification of WOs, synthesis of credible background textures, and relighting. Figure 5 gives an example of the effect of these steps, showing not only the original and repaired images, but also the results of background texture synthesis and computation of relighting. The smoothly-shaded white spots in the relighting image correspond to the recognized WOs.

3.2.1 How Is Automated Repair Different?

Fundamentally, what `DeOrbIt` does is very similar to the manual approach – although it also incorporates computation of lighting and relighting of the synthesized textures. The really difficult and different part is automatically identifying the WOs.

Perceptually, WOs seem very closely related to the out-of-focus (OOF) point spread function (PSF) images that make-up bokeh in an image. A perfect lens with a circular aperture generates an evenly-lit circular OOF PSF with a sharp edge – which sounds very much like the description of a WO. In fact, many X10 users mistake OOF PSF images for WOs. However, WOs differ significantly in that:

1. OOF PSFs have diameters related to distance from the focus point, whereas WOs have diameters related to how much charge is being leaked.
2. OOF PSFs for real lenses, such as the lens of the X10, usually have complex, repeatable, structures within them. These structures can include shadow/diffraction patterns from debris in/on the lens and shading variations revealing aspects of the optical design. The Fraunhofer diffraction central bright spot size is $2.44 \times \text{wavelength} \times (f/\text{number})$, which is larger than the X10's 2.2um sensel size even with the lens wide open. The result is that, for most aperture settings, the X10's OOF PSF is structured with the familiar concentric ring pattern of an Airy disc. This can be seen on the left side of Figure 3.
3. WOs always involve sensor saturation whereas OOF PSF often do not.
4. The edges of WOs frequently are “outlined” by darker-than-surrounding pixels. It is not clear what causes this.

With practice, it is fairly easy for a human to distinguish between OOF PSF and WO artifacts in images, most often by properties 2 or 3 above. Fundamentally, the algorithm used in `DeOrbIt` to identify the WOs is based on fragmenting saturated regions into pieces that are checked for saturated-circle properties which, fortunately, the X10's OOF PSFs usually do not share. It is theoretically possible to further improve rejection of PSFs as potential WOs by imposing a coded aperture upon the lens and using standard techniques to recognize the shaped PSF,⁸ but the need to modify the camera would not be an acceptable requirement for most X10 owners.

Unfortunately, WOs *overlapping* saturated regions that have not bloomed, or other WOs, often confuse the automated processing. Like many other small-sensor cameras, the FUJIFILM X10's limited dynamic range means it frequently captures images with many areas at saturation, and any portion of such a region easily could mimic whatever key properties an automated tool is using to distinguish WOs. Accepting this flaw, the tool is designed to err on the side of false positives; in this way, a user can easily overlay the original image with the tool's output and selectively erase the overly-aggressively de-orbed portions to reveal the original. Indeed, were de-orbing to be implemented inside an X10, it would make sense to still save both the original and de-orbed images to allow later editing.

Similarly, the tool allows not only the final de-orbed image to be output (e.g., the top right image of Figure 5), but also:

- The synthesized image without relighting (e.g., the bottom left image of Figure 5)
- The image of the relighting only (e.g., the bottom right image of Figure 5)
- Region edge distances (for debugging, hence not shown in the paper)
- The original unprocessed image (e.g., the top left image of Figure 5)

This provides great flexibility in manually applying repairs.

3.2.2 Overview Of The Processing

The latest `DeOrbIt` tool consists of a single source file containing just 2,133 lines of C code. The accompanying `Makefile` allows it to be built either as a stand-alone command-line program or as a CGI program using the `CGIC` library.¹ The command-line build is straightforward, processing files and writing the results. The CGI build is more complex. It actually logs each input image and parameter set, and then generates an entire HTML form to invoke itself. A portion of the HTML form describing the key parameters is given in Figure 6. The original and processed image URLs in the generated form actually link to the same program with the parameters

Processing Parameters

The `deorbit` processing parameters can be changed below. They are initialized to the values used to produce the image above. Note that it may be desirable to let `deorbit` be overly aggressive about processing regions and then selectively combine this output with the original by overlaying it in `gimp`, `photoshop`, etc., and manually erasing the portions that were too aggressively processed. The selection below facilitates even more aggressive tuning in your favorite image editor by giving access to the synthesized fill and relighting as separate images.

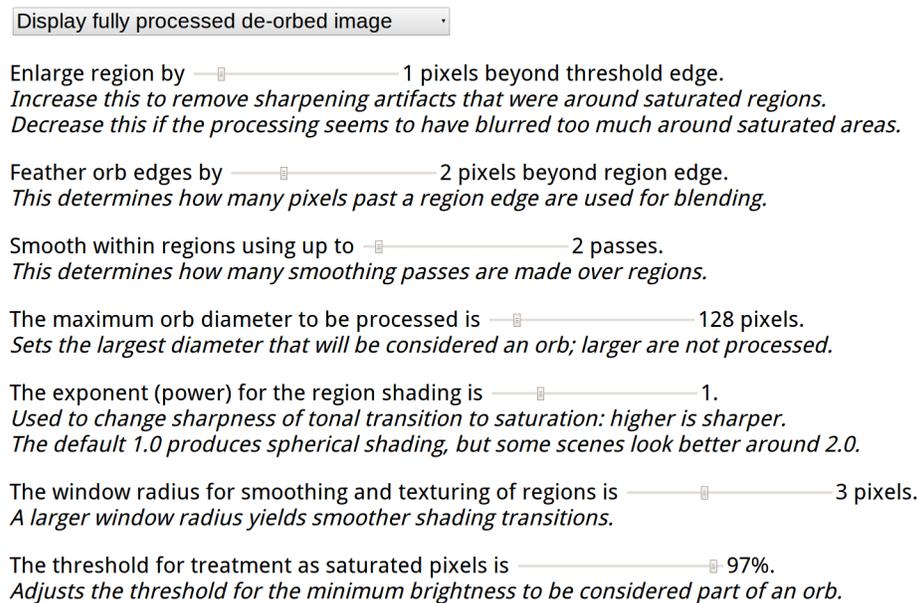


Figure 6. Processing parameters screenshot

set to fetch and process the desired logged image. Thus, the CGI version generates both the HTML form and the images in the form, but as separate HTTP requests.

After obtaining the argument/parameter values, the image is read and converted into a data structure containing an array of linear-gamma floating-point RGB values. There is nothing particularly special about this is done. The first few versions accepted PPM files to facilitate processing raw images using `dcraw`,³ but both released versions use `jpeglib` to accept ordinary JPEG images from the camera.

The key to making repair computationally cheap is to quickly limit the number of pixels being examined. The function `findwhite()` is responsible for locating all pixels that are potentially members of WOs and grouping them by which WO they belong to. It begins by simply marking all saturated pixels as their own groups. Because WOs are often surrounded by abnormally dark borders, the marked regions are then grown by a user-settable number of pixels to include the outlining pixels. This leaves each pixel marked as its own group. A non-recursive seed-fill algorithm is then used to unite adjacent pixels in groups, relabeling each pixel with the location of the first pixel in its group. The size of each group, in pixels, is also computed.

The next step involves computing the distance from each pixel to the nearest group edge. The algorithm used by `distmap()` constructs a memory-address-ordered work list of pixels to be labeled. Pixels outside of any group are initialized to 0 and omitted from the list. The algorithm then iterates removing pixels from the list as their distance labels become known. The iteration essentially reverses the list in each pass, thus flipping between increasing and decreasing address order – and optimizing cache line reuse.

Given that pixels are marked with their distance to a group edge, `doreach()` simply marks all members of each group with the highest distance that is reachable by a monotonic sequence of edge-distance increases from that pixel within that group. This is not necessarily the same value for all pixels within a group. For example, two partially overlapping WOs of different diameters typically have their local maximum distances separated by pixels that are nearer to an edge, thus, there is no monotonic path between the two maxima, and the two WOs

within the group are distinguished. The reaching algorithm used is non-recursive, first sorting active pixels into decreasing distance order and then using a single scan to propagate the reaching downward.

The reaching information serves several purposes. Another function, `dorsize()`, counts how many pixels are reached from each point. The roughly circular shape of orbs should mean that the ratio between the maximum distance and the count approximates the ratio between the radius and area of a circle. Yet another function, `domaxdist()`, determines the maximum distance for each group.

The basic re-lighting begins with `dolite()`. This function marks pixels that are within roughly circular shapes with a maximum radius determined by the distance marking. In other words, it distinguishes between the WO and un-bloomed portions of a saturated region. Conveniently, the OOF PSF of the X10's lens usually has a pattern which is not sufficiently similar to confuse this check.

Synthesis of credible background textures is done by `dosynth()`. The `gimp` resynthesizer is significantly more sophisticated – and slower. In fact, the resynthesizer alone generally takes longer to run than the entire `DeOrbIt` tool. The method used by `DeOrbIt` is simply to sort the active pixels in order of increasing distance from an edge and then propagate a randomly-biased summation from the already-colored pixels in a small neighborhood of each pixel. The textures generated by this process are not of particularly high quality for general use, but have a general tendency toward creating patterns converging toward the center of each WO. This mimics typical lens flare effects while still providing a smooth join of any surrounding textures. In truth, only a suggestion of this pattern will be visible through the relighting. Non-recursive, cache-aware, processing of the active pixels is performed by `dosmooth()` to apply a randomly-biased average from a small neighborhood around each pixel to smooth the synthesized background textures and better-integrate their edges with the surroundings.

The actual creation of the relighting “spot” image is done by `dospot()`, `dospotpaint()`, and `dospotblend()`. Shading is done based on distances from edges combined with “circle occupancy” tests, and uses a formula with a user-settable power to control the steepness of the lighting mask generated. However, this results in shading that looks too artificial. Thus, additional passes are made over the lighting image to smooth and randomize it, much as was done for the synthesized texture. The actual relighting is done by applying the formula `value.color=(texture.color + ((1.0 - texture.color) * relighting))`; in each color channel.

A basic premise is that the relit texture should never be brighter than the corresponding point in the original image, and this is ensured by `darken()`. If feathering is enabled, a final pass is made to feather and blend the resulting image using `domask()` and `doblend()`. The final image is restored to a standard gamma of 2.2 and output as a JPEG.

4. EVALUATION AND INSIGHTS

This paper gives several examples showing how `DeOrbIt` processed images that were captured as part of this study. For example, Figure 7 shows how a scene involving many specular reflections off the rippling surface of a pond was repaired. However, evaluating only images captured for the purpose of evaluating `DeOrbIt` could impose a bias, so real user images were collected and evaluated.

`DeOrbIt` was initially released as a tool with WWW form interface. The tool, currently in its second release, continues to be freely available via <http://aggregate.org/DIT/DEORBIT/>. This tool explicitly logs user images, processing parameters, user assessments of the results, and optionally the user email address (for the sole purpose of obtaining permission to use the photo submitted in a research publication). Permission was obtained via email for publication here of the images shown in Figures 8-11. Figure 8 is an example of an image that would have been spoiled by WOs, but was successfully automatically repaired by `DeOrbIt`. Large WOs over high-contrast scene detail make texture synthesis flaws more visible in the user image crop in Figure 9, but the automatic repair is still viable. Depending on parameter choices, the automatic repair of Figure 10 either leaves too sharp an edge on the huge WOs or causes the smaller window reflections to lose their shape (as shown); it would take either improvements to `DeOrbIt` or manual merging of repaired images to produce a higher-quality result. The huge saturated areas overlapping WOs confuse `DeOrbIt` sufficiently to make some regions of Figure 11 over/under processed no matter what parameters are used.

Users have applied the logging WWW tool hundreds of times, mostly before FUJIFILM offered to replace the sensor in affected cameras as a free repair. Thus, a variety of interesting issues were exposed.

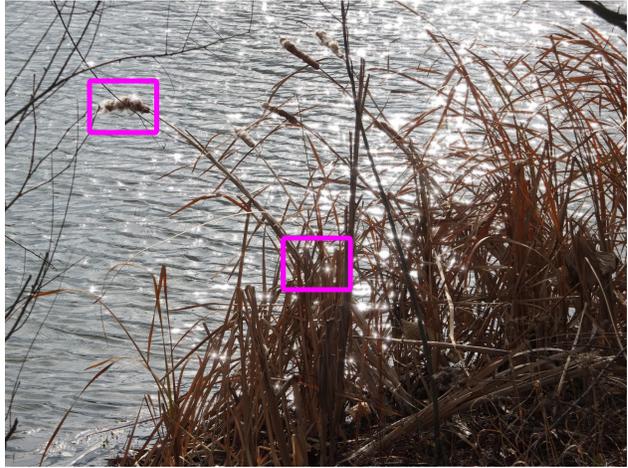
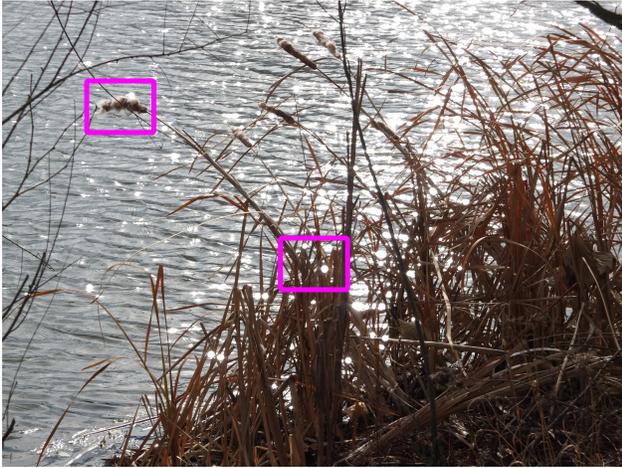


Figure 7. Pond Reeds: Left: Out-Of-Camera JPEG, Right: JPEG From De0rbIt

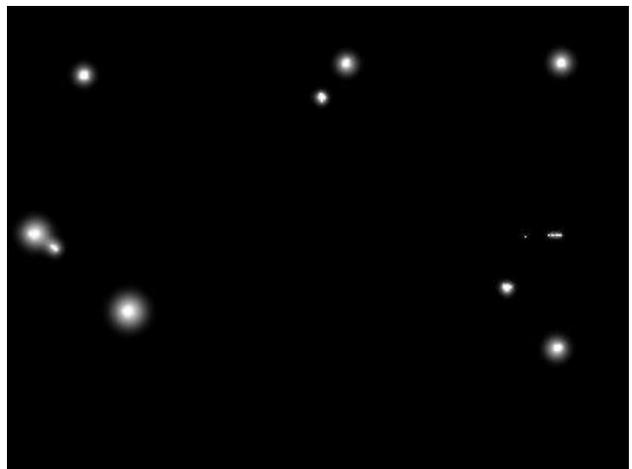


Figure 8. Millenium Bridge (by Paul Till): Original, Repaired, Cropped “ ”, Synthesized Texture, Lighting



Figure 9. Crop From Submitted Image (by M. Djebbari): Original, Automatically Repaired

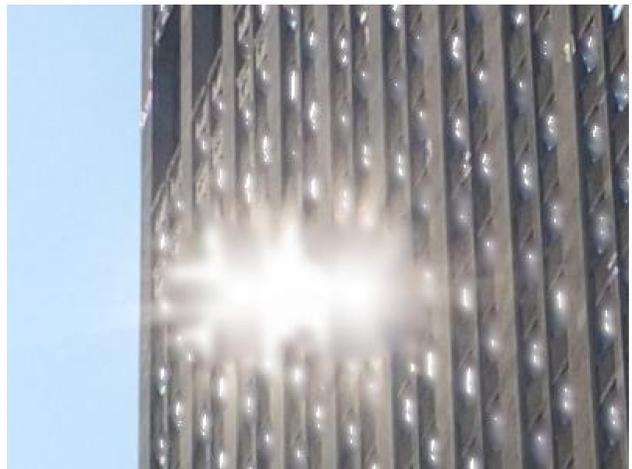


Figure 10. Crop From Submitted Image (by Victor Zaveduk): Original, Automatically Repaired



Figure 11. Crop From Submitted Image (by wymjym@wymjym.com): Original, Automatically Repaired

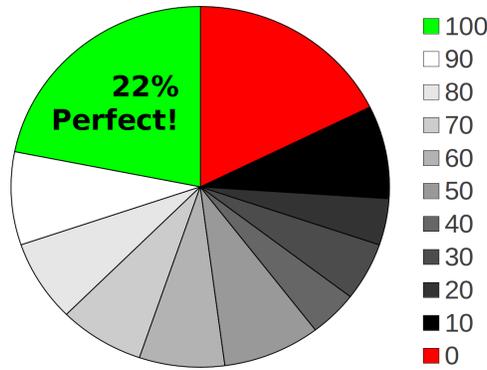


Figure 12. Distribution Of De0rbIt Repair Quality Scores

4.1 Where Do WOs Come From?

WOs are most likely to be caused by either specular reflections or bright light sources within the frame.

The most common source of WOs was specular reflections off of motor vehicles (especially cars and motorcycles), which accounted for 41% of all the images submitted. 32% of the images submitted, including quite a few of the shots of cars, were taken outdoors at night using streetlights as the primary light sources. Some of the worst-looking WOs scenes involved reflections off bodies of water, but these were not a large fraction of all images submitted.

4.2 Are WOs Really The Problem?

Of the images submitted to De0rbIt and logged, a surprisingly high 8% *did not contain any WOs*. Although some images contained a multitude of WOs, most contained only a few small WOs – and those few WOs might not have been the reason the user found the image in need of repair. There is supporting evidence of this in that the bulk of very low quality scores recorded were recorded for images in which De0rbIt *correctly did not process a non-orb saturated region*.

There is an interesting lesson in this. The WO problem is a serious image quality defect, but the truth seems to be that people were primarily upset about FUJIFILM’s poor handling of a reported defect – not upset by the defect itself. In fact, there are many users commenting online that they have never seen a WO from their original-sensor X10.

4.3 How Did De0rbIt Score?

The De0rbIt tool provides a simple slider to score the quality of its fixes:

Optional: You can grade the image quality of the reprocessed image above, which may help us improve the tool. Grades range from 0 to 100%. Your rating of the processed image quality is

Given the lack of any competing repair software, this seemed a reasonable way to get feedback on image quality. In retrospect, it left no calibrated basis for comparison. Perhaps it would have been more meaningful to ask which was better, the original or the repaired version of each image?

However, there are two additional issues that make determining a real user quality evaluation difficult. One is that, as mentioned before, there were many very low scores recorded for correctly not repairing images that did not contain WOs. The other is that scoring was entirely optional, and parameter-dependent. Some users tried multiple parameters and thus recorded multiple scores for processing the same image.

Figure 12 summarizes the votes and shows a very dramatic split. An impressively high 22% scored De0rbIt as having made a perfect repair. However, nearly 40% of the scores were below 50 on a scale from 0 to 100. Legitimately weaker scores are usually associated with issues like those seen in Figures 9-11. Comments in emails and public forums also revealed that some people find De0rbIt’s shading a little artificial because it is “too smooth” and hence looks “plastic” – despite De0rbIt’s deliberate introduction of random noise.

5. CONCLUSION

Given FUJIFILM's completely ineffective attempt at a firmware fix for the X10 WOs,² it was not initially clear that a software fix would be possible. However, `DeOrbIt` has provided proof that, at least in many cases, this type of camera system flaw can be largely hidden by computational means.

The new sensor loses the data for fewer pixels, which is clearly better than synthetic repair like `DeOrbIt` implements, but `DeOrbIt` remains a good answer for rescue of photos that already contain WOs. Where the fixes automatically applied are overly aggressive, it is easy to use the over-processed version as a layer to aid in manual repair of images. The tool itself certainly could do better, but further development has slowed to a halt since the new sensor became available. As a service to both the X10 user and image-processing research communities, `DeOrbIt` will continue to be freely available via: <http://aggregate.org/DIT/DEORBIT>

The WO problem also exposed some of the strange ways in which people react to technical problems. There were X10 users who did not notice the WOs, while others hallucinated WOs that were not present. The real problem seems to have been perceptions about FUJIFILM's handling of the fact that there was a flaw in their product. In truth, FUJIFILM came out with the new sensor, and offered free replacement, less than half a year after the problem was publically identified – that is a very short time for a design cycle. Continuing user complaints center on the fact that FUJIFILM has not recalled cameras for sensor replacement nor marked new camera packaging in a way that would allow a consumer to determine which version of the sensor is in a camera they are about to purchase.

REFERENCES

1. Thomas Boutell. `cgic`: an ANSI C library for CGI Programming. <http://www.boutell.com/cgic/>, 2011.
2. Barney Britton and Kelcey Smith. Fujifilm X10 'Orbs' Investigated. Does the Firmware Fix Work? <http://www.dpreview.com/articles/3340468423/fujifilm-x10-orbs-investigated-does-the-firmware-fix-work>, February 2012.
3. Dave Coffin. Decoding raw digital photos in Linux. <http://www.cybercom.net/~dcoffin/dcrow/>, 2010.
4. FUJIFILM. Statement Regarding the FUJIFILM X10 and X-S1 Digital Camera 'Blooming' Effect. <http://www.dpreview.com/news/2012/03/12/Fujifil-X10-X-S1-white-disc-cause-and-fixes>, March 2012.
5. FUJIFILM. X10 Firmware Update Ver. 2.00). http://www.fujifilm.com/support/digital_cameras/software/firmware/x/x10/, November 2012.
6. Paul Francis Harrison. Image Texture Tools: Texture Synthesis, Texture Transfer, and Plausible Restoration. *Ph.D. Dissertation*, 2005.
7. Paul Francis Harrison. Heal selection (fix to: Smart remove selection). <http://registry.gimp.org/node/15118>, March 2009.
8. Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 70, New York, NY, USA, 2007. ACM.