RECOVERING ANGLO-SAXON ERASURES:

SOME QUESTIONS, TOOLS AND TECHNIQUES

Peter A. Stokes

The "virtual" restoration of manuscripts by use of computers has received a good deal of attention in recent years. Perhaps best known is the so-called Archimedes Palimpsest, but other high-profile cases include the Dead Sea Scrolls, the Herculaneum papyri, and *Codex Sinaiticus*, all of which have been the subject and object of an extraordinary amount of highly specialised and hugely expensive research, as well as extensive media coverage.[1] However, using technology to recover text from damaged manuscripts need not necessarily require such effort and money, and much smaller projects are taking place all the time with relatively little financial outlay.[2] Such work does require some skill and practice, and no amount of skill is sufficient to recover material from the most severely damaged cases, but sufficiently good results can often be obtained with an average desktop computer and some readily available software. Given the relative ease with which this can be done, it is worth attempting such an approach before considering more expensive options and certainly before giving up entirely. The primary purpose of this paper, then, is to illustrate how otherwise lost readings can possibly be recovered by using high-resolution but otherwise unremarkable digital photographs of the sort which are now readily obtainable from many manuscript libraries. Not all possible types of damage can be considered here, and so the focus is on writing which has faded or been erased, possibly also having been overwritten at a later time, but the principles can be applied to many other situations as well.

Palimpsested, Erased and Damaged Manuscripts in Anglo-Saxon England

To begin, it is worth looking briefly at the surviving material from Anglo-Saxon England in terms of difficult or illegible readings and to ask how they differ with an eye to potential

[1] For a small sample of the literature on these manuscripts and their recovery see Easton; Easton and Knox; Easton, Knox and Christens-Barry; Salerno, Tonazzini and Bedini; Netz and Noel.
[2] For two examples see Schipper *passim* and Craig-McFeely *passim*.

recovery. The first thing to note is that there are very few palimpsests which survive from Anglo-Saxon England. Neil Ker noted only one fragment containing Old English which is a "true" example and observed that this was palimpsested in Trier rather than England.[3] Lowe, with slightly looser criteria, gave three more entries and noted one more which he rejected from his list.[4] Despite this relative dearth there are still many examples of erasure and rewriting in Anglo-Saxon manuscripts.[5] Most of these are relatively straightforward corrections to individual words or letters, and recovering the original readings may be very useful to anyone trying to recover the process by which the manuscript was copied or compiled. Rather different in nature is the much smaller corpus of material which has been erased more fully, perhaps with a different text written over the top but without the entire page being cleaned off as is normally required for a palimpsest. These include most notably erased inscriptions which can give important evidence about provenance, and also the body of documents, including those in Old English, which have been erased from gospel-books or other liturgical manuscripts.[6] In addition to these cases resulting from deliberate erasure, we also have cases of apparently accidental wear with the result much like that of erasure. Examples here include the copy of Solomon and Saturn which is now in pages 1–26 of Cambridge, Corpus Christi College 422, and the endorsement (perhaps erased or more likely heavily rubbed) of BL Cotton Augustus ii. 6, a single-sheet charter from Pershore.[7] Similar in the category of accidental loss are some inks which have faded very badly, to the point of illegibility. Fortunately the iron-gall inks used by the Anglo-Saxons were normally quite dark and also quite chemically stable, thus the manuscripts are generally free from fading or

[3] Ker lxii, referring to Rome, BAV reg. lat. 497, fol. 71 (his no. 391).

[4] Lowe, "Codices Rescripti", 72, 76 n. 13, and nos 10, 23, 35, and 64. The rejection is Durham, Cathedral Library A. ii. 16, fols. 1–23, 34–86 and 102 + Cambridge, Magdalene College, Pepys 2981 (18); part of this manuscript was erased but rewritten by the same scribe; Lowe, *CLA* 2, no. 148a. For further discussion of Anglo-Saxon palimpsests see also now Papahagi's and Szarmach's contributions to this volume.

[5] See especially Szarmach's contribution to this volume, pp. 1–4.

[6] For some of these see Ker nos. 6a; 35, 1–3 + 8; 119b; 126; 147a; 185f; 194 (but see below); 249a, f; 291d (largely illegible but not necessarily erased); 370b–c. Ker's no. 194 is BL Cotton Tiberius B. v, vol. 1, fol. 75: this was described by Ker as containing records added in blank spaces (p. 256) but was listed as a palimpsest by Lowe, "Codices Rescripti", no. 35. Lowe later described the page in more detail, noting that it has documentary additions both in space originally left blank and also on erased portions of text: Lowe, *CLA* 2, no. 190. The page was not described or illustrated in the printed facsimile of the volume edited by McGurk because it was added in the sixteenth century by Robert Cotton.

[7] A complete digital facsimile of CCCC 422 is available at *Parker on the Web*, and photographs of the face and dorse of BL Cotton Augustus ii. 6 as Keynes no. 208.

corrosion. However, the red inks used for rubrics were not always so stable, and in some cases they have faded quite significantly.[8]

Clean erasures, rubbing, and faded ink are usually relatively easy to recover. This assumes that the parchment is fairly clean and uniform in the region of the illegible text, and that at least some of the text remains and that the difference in colour between the illegible writing and the parchment is still large enough for a digital camera to capture it. Rather more complicated, and also more common in Anglo-Saxon material, is when the illegible writing is obscured by other ink on top, as is often the case with corrections or obliterated inscriptions. Similar but complex in somewhat different ways are those where the damage has resulted not so much in faded ink as darkened parchment. If the parchment is uniform in discolouration then it again can usually be enhanced without too much difficulty. In practice this is rarely the case, however. Indeed, perhaps the largest body of illegible script comes from the library of Robert Cotton. This now forms the Cotton collection in the British Library and is the single largest repository of material in Old English and indeed of Anglo-Saxon manuscripts in general.[9] Unfortunately the books were badly damaged in a fire in 1731, as a result of which many of the pages are now burnt, shrunken, split, dirty, darkened by fire or water, or any combination of these.[10]

A SHORT INTRODUCTION TO COMPUTER IMAGING

This is the problem, one which has been encountered by almost everyone who works with medieval (or even modern) manuscripts. But what are the possible solutions? Fortunately, there is now a lot that can be done very cheaply and easily in this age of powerful and easily accessible computers and with the relative availability of digital photographs. Most of the remainder of this paper will therefore consider some of the ways in which lost and particularly erased readings can be restored. The field of digital image-enhancement is very large and very active, and even a

[8] For example, see many of the rubrics in the so-called "Red Book of Darley", now CCCC 422, pp. 26–586. See Budny 1:650–51 (no. 44) and note 7, above.

[9] The Cotton collection now holds 123 distinct items from Anglo-Saxon England, of which 99 contain material in Old English. The next largest collection in both regards is that of Corpus Christi, Cambridge, which holds 84 and 53 items respectively. These counts are based on my own data compiled from Gneuss' handlist and Ker's catalogue; for a discussion of the method involved and the underlying assumptions underlying it see Stokes 1:3–4. Compare also Ker liv for a list of manuscripts and *membra disiecta* which Cotton owned and which contain Old English.

[10] Prescott *passim*.

brief survey of the field is far beyond the scope of this article. Instead of attempting to do this, therefore, it seems more productive to consider some relatively basic but still proven techniques which were developed through practical experience, particularly that gained while working for the British Library on a project to digitally restore their entire collection of Greek palimpsests.[11] None of these techniques require any special photography, computer hardware, or rare or expensive software. In order to use them effectively, however, one must first understand at least a few basics of computer imaging.

The first step in any enhancement is to obtain a digital photograph of the manuscript in question. Although often underestimated, this is by no means trivial and needs to be done with some care. Indeed, it is properly the subject of an entire paper by itself, and fortunately it has already been treated very competently by Julia Craig-McFeely.[12]

Let us assume that there is available a high-quality digital photograph taken in natural light in a standard format such as TIFF. This photograph, like any digital image, is in essence a grid of coloured points. Each element in the grid is known as a "picture element", or "pixel", and each pixel is assigned to a particular colour in order to create an image. The principle is straightforward, but the details are rather more complex. Particularly relevant here is the question of colour. In terms of physics, colour is simply a way of sensing different wavelengths of visible light. At the risk of oversimplifying both quantum mechanics and human psychology, light can be thought of as waves in an electromagnetic field, and as the wave oscillates more or less quickly, so we perceive this wave as different colours. Thus slower waves look red to us, slightly faster ones look orange, and so on through the entire spectrum, going via yellow, green, and blue in that order. These wavelengths are then sensed by optic nerves and interpreted by the brain as colour, and indeed most people can distinguish millions of different colours without any effort.

Such is the case for people, but computers work rather differently. The problem is how to represent colour inside a computer, in such a way that the computer can store, manipulate, and ultimately reproduce what is in essence a physical, if not psychological, phenomenon. Ultimately, computers are designed in such a way that they can only work with one type of

[11] *Rinascimento virtuale - Digitale Palimpsestforschung.*

[12] Craig-McFeely §§13–47; Craig-McFeely and Lock 12–16.

information – numbers – and therefore every piece of data that we want to put into a computer must somehow be encoded as a number. To phrase the problem slightly differently, then, engineers need a way to take a specific wavelength of visible light and represent that as a number, and conversely to take a number and represent that as a specific wavelength of light. In practice, this is exactly what scanners and digital cameras do on the one hand, and printers and monitors do on the other. Scanners and digital cameras contain sensors which measure the wavelengths of light and represent those as numbers for processing and storing in a computer. Monitors take those stored numbers and convert them back into visible light. Printers also convert numbers into colours but in a different way: rather than generating light directly, they produce inks which absorb light that is already there in the room. This difference has some important implications but fortunately these implications are of little immediate consequence for digital image-enhancement and so they shall not be considered here.[13]

How does all this work in practice? There are many different ways of storing colours and images.[14] When displaying images on a monitor the same principle is almost always used: the so-called RGB, or "red-green-blue" colour system. This principle is very simple, namely that every colour which can be displayed on a monitor is represented by three different numbers, with these numbers representing the amount of red light, green light, and blue light respectively. It may seem rather counter-intuitive, but it is possible to produce almost all visible colours simply by mixing different amounts of red, green and blue light. If this were not the case then it would be almost impossible to build monitors which could display more than just a few colours, as millions of differently coloured lights would need to be produced in every monitor and television, rather than three different light-sources as are used in practice.

The reason why only three colours are necessary is based on the biology of human vision. Specifically, human eyes have three different colour sensors which detect light at different frequencies. One sensor is most sensitive to light which is at the red end of the spectrum, one is most sensitive to light towards the blue end, and the third, the most sensitive, is active around the

[13] The most obvious consequence of this difference is that our perception of a printed page changes significantly in different light, whereas that of a computer monitor is much less affected. To take an extreme example, if a room is completely dark then a computer monitor can still be seen, whereas a printed page is entirely invisible. For a further consequence, the difference between additive and subtractive colour-mixing, see below, note 15.
[14] For a brief overview see Craig-McFeely and Lock 12–15.

green region. This principle is shown in Figure 1, below, where the curves reflect the relative sensitivities of the three sensors at different wavelengths. Yellow light, for example, causes a response of a certain intensity in the “green” sensor and also a response of another intensity in the “red” sensor, and our brain mixes these two responses together and interprets them as yellow. However, the important point here is that the same two responses can be reproduced by using two different lights of different colour and intensity, neither of which is yellow, and shining these two lights together onto the same sensors in our eye. These two different lights, if correctly chosen, will stimulate these two sensors exactly as a single yellow light would have done, and thus the brain interprets these two lights as one. This, then, is the secret of colour displays, and if one looks very closely at an older television screen or very old computer monitor then the three differently coloured lights are visible to the naked eye.

How does this relate to the storage of colour in a computer? As noted above, most modern computers represent the colour of each pixel with three numbers, these numbers in turn representing the intensities of the red, green, and blue lights which are required to produce the colour in question. These intensities are usually encoded using a whole number from zero to 255 (inclusive), so zero is “off”, no light at all, and 255 is “on full”, or maximum brightness. This then gives us 256 different intensities for each of the three lights, and thus 256×256×256 = 16,777,216 different colours in total. Note that this is a maximum number of colours which can be represented internally; the number which can be displayed on a monitor is less and depends on the quality of the monitor and how well it is configured. With this system the basic colours, red, green and blue, can be represented by (255, 0, 0), (0, 255, 0), and (0, 0, 255) respectively. Mixing two of these colours in each of the three possible combinations gives cyan, magenta, and yellow, as (0, 255, 255), (255, 0, 255), and (255, 255, 0) respectively.[15] Black is produced by showing no light at all, as one would expect, and so all three values are set to zero. White is somewhat less intuitive: it is the result of all three sensors in the eye receiving maximum value, and this translates to having all three lights on full, so (255, 255, 255). Greys are achieved by mixing the three colours in equal amounts; thus (100, 100, 100) is a dark grey, (127, 127, 127) is

[15] It should be noted that the mixing here, namely mixing light, is additive, as discussed above, note 13. The results are therefore quite different from those obtained by mixing paint, which is subtractive, and where the primary colours are red, yellow and blue rather than red, green and blue.

mid-grey, and (200, 200, 200) is a light grey. Note that two of the three values are therefore redundant in these cases and so an entire image that is purely grey requires only one number for each pixel; this is a point which will become important shortly.

IMAGE PROCESSING: TECHNIQUES

Now that a few basic points have been considered regarding image-processing, it remains to be seen how these can be used. There are many different techniques and principles, of course, and only a very small selection can be considered here. Specifically, two important principles of image-manipulation will be considered here: adjusting histograms and manipulating colour-channels.

Histograms

One of the most important principles in image-processing is the histogram. Histograms are widely used in statistics and in principle simply represent counts of items; a trivial example is shown in Figure 2. In image-processing, the histogram of an image shows the number of pixels in that image which have been set to a particular colour. Normally the histograms are separated into each of the three colour channels for ease of representation. For example, take a plain yellow square. As discussed above, yellow is formed by a mixture of red and green light, and thus in a digital image of this square most of the pixels are very bright in the red and green channels and very dark in the blue channel. This is reflected in the three histograms for that image, shown in Figure 3. Another example is shown in Figure 4: this time many of the pixels are bright blue and without much red or green, and this can easily be explained by the bright blue sky in the photograph. There is also some medium-intensity yellow in the sand, and some dark colours in the bushes, and these are also reflected in the peaks in the middle of the red and green histograms and at the lower end of all three histograms.

Histograms, then, are useful as a quick statistical summary of an image and the distribution of colours within it. They can also be used to manipulate the image by altering those statistics. In particular, the histogram reflects the range of different colours in an image, and by manipulating this we can improve the visibility of that image. For example, consider a very faint portion of text on an otherwise white parchment. The question is how to recover this erasure, and this depends in turn on what "very faint" means to a computer. "Very faint" really means that the writing is almost the same colour as the parchment and therefore that there is too little difference between the writing and the parchment for the human eye to be able to see the letters. We can see this by creating an artificial "erasure" by writing faint text on top of a coloured background and then looking at the resulting histograms, as show in Figure 5. The narrow peaks in the histograms show that there is a very small range of different colours in the image and furthermore that all of these colours are very close to each other. However, as noted above, the average computer display can show many millions of colours, and so one way of improving the image is to spread out the colours in the image, using more of the available range and thereby increasing the difference between writing and background. Fortunately, we can do this very easily with a computer.

This spreading of histograms, like many other techniques of image enhancement, can be achieved with a variety of different pieces of software. I shall confine my discussion to the two which are widely used and readily accessible: Adobe Photoshop CS and the GNU Image Manipulation Program, or GIMP.[16] Photoshop is commercially available and must be purchased but is already present on many computers, particularly in academic systems. The GIMP, in contrast, is published under the GNU Public Licence, or GPL, and so it can be downloaded from the internet for free.[17] Both pieces of software are available for Windows and Mac OS X systems, and the GIMP is also available for Linux.

The process of spreading out the colours of our faint writing is the same in both Photoshop and the GIMP, and in both cases it is called "Level Adjust". Selecting this command from the relevant menu brings up a dialogue-box on the screen and that box in turn contains a histogram of the image and various controls to manipulate it. The full details of this are rather

[16] "Adobe" and "Photoshop" are registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Adobe Photoshop CS is hereafter referred to as Photoshop. GIMP is copyright under the GNU General Public Licence (GPL): see *GIMP*.

[17] "GNU General Public Licence".

complex, but further information can be found in the user-manuals for the relevant software. The key to understanding what needs to be done is to look at the histogram. In some cases there will be a single peak which is fairly narrow, and the rest of the histogram will be low and flat. This narrow peak confirms that most of the pixels in the image are the same colour and therefore that it is difficult to distinguish different parts of the image. In other cases there may be two distinct peaks, for example in a palimpsest with dark overwriting, and faint underwriting which is approximately the same colour as the parchment. Another possibility is one somewhat wider peak with two sub-peaks; this might be the result of writing which is faint but still distinguishable from the background. All of these cases are illustrated in Figure 6.

In order to enhance the image, the range of colours needs to be increased. In other words, the lightest colours in the image need to be made very light, and the darkest colours very dark. For example, all of the pixels in a given channel in the image might have intensities between 100 and 150. In this case, those pixels in that channel with intensity of 100 should be reassigned to an intensity of zero, and those of 150 to an intensity of 255, with everything in between being spread out evenly across the full range of 0–255. This makes the darkest pixels black, the lightest ones white, and it increases the difference in colour between all the pixels in the image in that channel. To do this in Photoshop or the GIMP, we therefore need to specify the range of intensities below which all pixels should be assigned to black, and similarly the range above which the pixels should become maximum. This is achieved by two small arrows which are visible at the bottom of the appropriate histogram. Initially one arrow is pointing to the very bottom of the histogram, and the other is pointing to the very top; thus the lowest possible value, 0, is black, and the highest, 255, is full intensity. As we move the lower pointer up, more and more of the darker colours are assigned to black, and the intermediate colours are spread out further and further across the resulting range. The equivalent then occurs as we move the upper pointer down. In most cases, the lower pointer should be positioned at the point where the large peak just begins to grow, and similarly the upper pointer should be positioned at the other side of the peak. Because this is such a common requirement, most software packages have an "auto" button on the level-adjust; clicking on this automatically positions the pointers at the point which the computer thinks will be most useful. This is useful as a starting guess and can then be adjusted manually. The software also usually allows adjusting different channels individually, or adjusting the combined intensity of all the channels at once.

This very simple technique can be remarkably effective with faded writing and is also very quick to do. It does have one significant limitation, namely that it enhances everything together without any easy way of discriminating between what is desired and what is not. It is therefore most effective when the parchment is very clean and smooth, since otherwise even faint blemishes such as hair follicles will become clear in exactly the same way that the faint writing does. In practice this means that it is most useful for clean erasures or ink which has naturally faded on otherwise good parchment. It is generally much less useful for dirty parchment, including that which has been blackened by fire, and it normally works better with the flesh side of parchment than the hair side. Even with these limitations it is still a quick and simple technique which can be very effective, and so it is usually worth trying it as a first step.

Channel Manipulation

The other approach to be considered here is somewhat more complex, but it overcomes some of the difficulties of the simple Level Adjust. This second approach uses the computer's representation of colour to our advantage. As has been discussed above, every pixel in a colour image can be represented by three numbers. This then raises the possibility of adding, subtracting, or otherwise manipulating these numbers inside a computer. In particular, we could calculate a "weighted average" of the three values for every pixel in the entire image. As one example, we could take 30% of the red channel, 60% of the green channel, and 10% of the blue channel, and add the three values together to produce a single number. This may seem like a pointless exercise, but it is instead extremely useful. This is because, as has been discussed above, if each pixel has a single value then the image can still be displayed, but in shades of grey instead of in colour. If the proportions of each channel are chosen correctly then the results can be striking.

To illustrate, let us consider an ideal, artificial example. Imagine first that the parchment is pure white, and that it has some dirt on it which is grey. Let us also imagine that we have a palimpsest where the overwriting, that is the top writing which we want to get rid of, is entirely black. Finally, let the underwriting which we want to recover be bright red. In this case, adjusting the levels has little benefit as it would enhance the overwriting and dirt just as much as it would the underwriting. To overcome this, the colour-channels need to be manipulated in such a way as to remove the noise and overwriting but to leave the underwriting intact. Specifically, if we subtract the blue channel from the red channel while ignoring the green channel entirely, we obtain the values shown in Table 1. The numbers in the Red, Green, and Blue columns give the values in each of those channels for the four different materials in the image, and the fourth column gives the result of the manipulation. If we display the result as a greyscale image then all of the noise, overwriting, and parchment will be black, and the underwriting is left on its own as entirely white.

Item	Red	Green	Blue	Red column minus Blue column
Parchment (white)	255	255	255	0 (black)
Noise (grey)	122	122	122	0 (black)
Overwriting (black)	0	0	0	0 (black)
Underwriting (red)	255	0	0	255 (white)

Table 1: Colour values of artificial "palimpsest"

The example just given is clearly ideal and it is unreasonable to expect such results in practice. Nevertheless, the principle still holds with more complex examples. If the parchment is still white and the dirt still grey, but the overwriting and underwriting are more similar to each other in colour, and the underwriting is quite faint, then the intensities might be something like those shown in Table 2.

Item	Red	Green	Blue	Red column minus 84% of Green column
Parchment (white)	255	255	255	41 (dark grey)
Noise (grey)	238	238	238	38 (dark grey)
Overwriting (faint red)	151	135	135	38 (dark grey)
Underwriting (faint pink)	243	219	219	60 (lighter grey)

Table 2: Colour values of second artificial "palimpsest"

In this case, we need to subtract only a portion of the green channel, 84% to be precise, as again shown in Table 2. Once again the resulting image has parchment, dirt and overwriting all approximately the same value, and the underwriting significantly different. The underwriting is still quite dark, however, as is apparent from the figures in Table 2. Fortunately this is again the problem of faint script, and the solution is simply to adjust the levels, as discussed above. The results of this are shown in Figure 7, with the first image being the original before any enhancement, and the second the result of manipulation as just described.

The question which remains is how to perform these manipulations in practice. The relative weightings of the three channels can be obtained by a series of mathematical calculations, but fortunately this is not normally required.[18] Instead, many programmes such as Photoshop or the GIMP allow the user to adjust the weightings of the three channels and to immediately see the results; this allows good results to be obtained relatively quickly just by trial and error. The menu-items to do this vary slightly but are the same in principle. With Photoshop, for example, it is performed using the "Calculations…" command in the "Image" menu. In the GIMP it is "Channel Mixer…" which is found under "Components" in the "Colours" menu.[19] Again, the details vary, but usually a dialogue-box is presented with three sliders, one for each channel, and these can be adjusted to give the different weights. In some cases one can choose between colour and monochrome output; in this case monochrome is normally the more useful. It is often also possible to "preserve luminosity" but this is rarely useful in practice. It can be helpful to turn this feature on while the weights of the different channels are still being adjusted, but it should normally be turned off again before the dialogue-box is closed and the settings accepted.

Combining Techniques using Layers

Two different techniques for enhancing images have been presented so far. How, then, can they be used in combination for a particular image? There is no single answer as every case has its

[18] For one method of performing such calculations see Easton, slides 32–43, and for a more complex approach see Salerno, Tonazzini and Bedini *passim.*

[19] "Colors" is a primary menu in the GIMP version 2.4. In earlier versions it was a submenu under "Layers", but the "Decompose" function was in the "Color Filters" submenu of the "Filters" menu.

own challenges. However, experience has shown that a particular sequence of steps is often effective. Specifically, these steps are as follows:

1. Blur the image.
2. Copy the image.
3. Manipulate the channels of one copy of the image, as described above.
4. Adjust the levels of this copy of the image, as described above.
5. Invert the adjusted copy of the image if necessary.
6. Paste the adjusted copy of the image back onto the original in a new layer.
7. Change the form of overlay to that which is most effective (usually one of "Normal", "Multiply", or "Difference").
8. Try adjusting the levels of the lower (original) image.
9. Flatten the resulting image.
10. Adjust the levels of the flattened image, as described above.

Steps 2–4, 8 and 10 are either described above or are simple operations which should be familiar to anyone who uses a computer. The remaining steps require further explanation.

The first step listed above is to blur the image. This may seem counterintuitive, as the final objective is to obtain a clear image. However, blurring an image normally reduces the impact of small, sudden changes of colour in an image and increases the impact of larger blocks of colour. This is often precisely what we want in an image, as the small sudden changes are often hair follicles, bits of dirt, and so on, and the larger blocks of colour are often ink and parchment. This in turn assumes a high-resolution image and it also depends on the blurring which is used, but often the writing is more legible after blurring. Blurring is a simple function which is present in programmes like Adobe Photoshop and the GIMP, and indeed most such programmes offer several forms of blur. In practice it usually makes little difference which is chosen and the simple "Blur" is normally sufficient.[20] A similar alternative is the "Despeckle" filter which is present in both Adobe Photoshop and the GIMP. This is designed to remove

[20] Both "Blur" and "Despeckle" are available under the "Filters" menu in both Photoshop and the GIMP. "Despeckle" is under the "Noise" submenu in Photoshop and the "Enhance" submenu in the GIMP.

small points of dirt and such without blurring other features such as the line between ink and parchment. As always, experimentation is required to see which is most effective.

Step 5 suggests inversion of the image. This depends on the relative weightings given to the channels in Step 4: very often the main script is white and the parchment black, as described in the examples of the previous section. If this is the case then the image can be inverted so that the script is black on a white background. Although this is not necessary to simply read the text, it is required for the following stages to be successful.

Steps 6–9 specify pasting the enhanced image back onto the original as a new layer. In software like Photoshop and the GIMP, images are treated as having layers which sit on top of one another. This is most obviously useful for people who are creating original artwork. If one wishes to put text onto a photograph, for example, then the photograph could be on one layer and the text on another. This then allows the artist to treat the two elements separately, to make one layer invisible in order to better work with the other, and so on. This can also be used for enhancing images but in a slightly different way. The procedure is to copy the enhanced image, then select the original image and paste on top of it. The details vary according to the programme, but normally the pasted image will be "floating" on top of the original and will obscure it entirely. However, there will also normally be a "Layers" palette or menu in the software, and this will have an option to "Create New Layer" from the pasted image. This then allows the user to show the lower, original image, the new image on top, or both. However, it also allows the combining of the two images in different ways by selecting different "modes". Experimentation will again reveal the most effective combination, but often "Normal", "Multiply" or "Difference" is the most effective.[21] It should also be noted that the relative opacity of the layers can also be controlled. Thus using "Normal" mode with the default opacity of 100% means that the top layer obscures the lower one entirely. However, as the opacity is reduced, so the lower layer becomes visible. Once again, experimentation here can yield very good results. Finally, when the result is satisfactory, the image can be "flattened". This is a simple operation which combines the different layers into one so that the result can be saved as a single

[21] For further details see Craig-McFeely and Lock 32–34 and 48–64 *passim*. This functionality is not unique to Photoshop but is also available in the GIMP, *pace* Craig-McFeely and Lock 32.

image in TIFF format or similar. In order to aid this process, a summary of some of the more useful functions in both Photoshop and the GIMP is given in Table 3.

Function	**Adobe Photoshop CS**	**GIMP**
Blur	Filter > Blur > Blur Filter > Noise > Despeckle	Filters > Blur > Blur Filters > Enhance > Despeckle
Combine channels	Image > Calculations...	Colors > Channel Mixer... *or* Colors > Decompose...
Level Adjust	Image > Adjustments > Levels...	Colors > Levels...
Rotate	Image > Rotate Canvas *or* Select All, then Edit > Transform > Rotate	Image > Transform > Rotate *or* "Image Rotate" Tool
Invert colours	Image > Adjustments > Invert	Colors > Invert
Show histogram	Window > Histogram	Colors > Histogram
Image Overlay	Copy & past image, then "Layers" Window	Copy & paste image, then "Layers" Window
Flatten Image	Layer > Flatten Image	Image > Flatten Image

Table 3: Summary of relevant functions in Photoshop and the GIMP
Note that in versions of the GIMP earlier than 2.4 Colors is a submenu under Layers, and both Decompose and Channel Mixer are in Filters > Color Filters.

THE IMAGE VIEWER

The discussion so far has provided a few basic tools which can be used to good effect when recovering illegible writing from medieval manuscripts. There are a series of steps which can be tried, and these are often sufficient for relatively many cases. Putting these steps into practice is by no means trivial, however. In particular, almost all of them allow some degree of freedom and therefore require some degree of experimentation in turn. One must decide whether, how, and by how much to blur the image; what weights to give the different channels; how much to adjust the levels; what mode of overlay to use; and so on. Unfortunately, the implications of each decision are not clear until the very end of the process. If the blur is not appropriate, then one must return to the very start and repeat the entire procedure, particularly since the previously obtained values for channel-weighting, level-adjustment and so on are potentially made invalid by the altered blur. This is a significant impediment in practice and is one of the biggest impediments to the rapid and effective enhancement of images. To avoid this difficulty I propose a different model for the manipulation of images. Instead of viewing the entire procedure as a sequence of discrete, one-off steps, with each having to be completed before the next is begun, instead it seems more useful to consider the whole process as a continuous flow. Specifically, it would be very useful for the person trying to recover the reading, if the final result of all the stages could be seen even as the settings for an early stage are being adjusted. There are at least two good reasons why software such as Photoshop and the GIMP do not do this. One is that it is entirely unnecessary for most people, as in most cases the output of a given step can immediately be seen to have been successful or otherwise. Furthermore, the alternative view of image-processing requires a great deal of computing-power since the entire sequence of operations must still be repeated every time a single value is changed; the only difference is that the computer does it automatically rather than requiring human intervention. Nevertheless, computers are become more and more powerful and are capable of carrying out such processing at greater and greater speeds. In many cases, moreover, the writing which a researcher wishes to recover is confined to a relatively small part of the page, in which case the image in question is similarly small and thus relatively fast to process. All of these factors suggest that a system such as this may be of value.

To test this principle, I have developed prototype software to implement such a system which incorporates the basic techniques discussed in this paper, along with one or two others.[22] The interface is arranged rather differently from conventional image-manipulation software: rather than being commands in menus, the various stages are represented by a sequential series of panels which runs down one side of the screen. This is illustrated in Figure 8. This may look more imposing at first, but it allows the researcher to immediately see exactly what the stages are in the processing, to see what settings have been used for each stage, and to adjust those settings with immediate effect. Although not yet possible, in principle the researcher should be able to change the order of these panels, and to add or delete panels as required. Indeed, almost all of the functionality in Photoshop or the GIMP could theoretically be implemented in this way,

[22] At the time of writing, the prototype implements channel manipulation, "Level Adjust" (both manual and automatic), rotation and reflection, a 3×3 convolve filter (which includes a "Blur" facility), thresholding (both manual and various types of automatic), invert, and zoom. All functionality was implemented using the Java Advanced Imaging (JAI) library. Java is a trademark or registered trademark of Sun Microsystems, Inc. or its subsidiaries in the United States and other countries. See *Java.com* and "Java Advanced Imaging". The software is still in early stages of development but will ultimately be freely available online under a Creative Commons Licence or similar. Anyone interested in a copy of the software before then can contact the author directly but with the understanding that it will not be reliable or complete.

although this would be limited by the time required for processing. Indeed, to overcome this limitation two further options have been added. The first is currently labelled "Live Updates": when selected it updates the image continuously with every adjustment; thus if a slider is moved then the image is processed for each of a series of intermediate values between the start and end-points of the slider. This means that the image changes before the eyes of the person moving the slider: a feature which is very useful when trying to establish the ideal setting but which can make the whole system very slow and unresponsive for large images or complex processes. Similarly, the user can select "Show Original Image", in which case the image is shown as it first was, before any enhancement. This serves two purposes: one is simply to compare "before" and "after" images and thereby to establish how much a given series of enhancements has improved the legibility of the text. The second is again for reasons of performance. If the researcher has a pretty good guess of what the best settings might be, then he or she may wish refrain from processing the image until the correct settings are all in place. This avoids all the problems of performance, as the image is not actually processed until the "Show Original Image" option is deselected. Thus even a very complex process can be set up without any impact on performance and then run once when the original image is no longer shown. In many respects this is like the model provided by Photoshop or the GIMP, in that the whole process is done once and so the user cannot see the effect of specific changes. The difference, however, is that the researcher with this new system can still alter values of early stages without repeating the entire process.

REPRODUCIBILITY AND THE ETHICS OF ENHANCEMENT

One fundamental issue of both practical and theoretical importance underlies all of this work on image enhancement for scholarly purposes: that of reproducibility, with the associated issues of documentation and accountability. Scholars are often suspicious of image-enhancement and for good reason: all manner of dishonesty can be perpetrated with a bit of practice. Furthermore, it is not unusual for the script of an enhanced image to be legible on a computer screen but difficult or impossible to capture in print. These issues raise the question of how one can verify claims of a particular reading. The answer, I think, is readily available in the sciences where each experimental procedure must be documented in such a way that it can be reproduced and verified by independent researchers; only after such verification can the result then be accepted by the scientific community. Translating this to the present situation, it follows that a very precise record must be kept of exactly what was done to any given image, so that anyone else can repeat the steps themselves and independently verify the results. Indeed, an international standard already exists for the recording of such information about an image, the very existence of such a standard highlighting its importance.[23]

Although such a standard exists, it is still very difficult to implement in practice for three reasons. The first is that the standard itself is very difficult to set up and it requires a very high level of technical understanding in order to make it useable for a given situation. This is not such a problem in a large-scale project (for which it is designed) as such projects normally have the required expertise available; it is a substantial difficulty for scholars working on their own individual research, however. The second difficulty is that simply recording the relevant data is tedious and very prone to error. Packages like Photoshop and the GIMP offer no easy facility to record operations in a way that they can be easily transferred to other systems. It therefore requires a great deal of discipline for anyone enhancing images to manually note every setting of every stage throughout the entire process, and the potential to forget this is very high, particularly when so much experimentation is required. The third difficulty is that the precise algorithms applied by proprietary software such as Photoshop are rarely made public. The standard requires noting not only what operation was carried out but also what software and what version of the software was used. This means that the operation can be repeated in principle, but it depends on the precise version of software still being available. The problem is somewhat alleviated with open-source projects like the GIMP because one can examine the internals of the software to see exactly what process was applied, and these details can then be included in the record of steps taken. However, examining complex computer code like this is

[23] The standard is ANSI/NISO Z39.87-2006, for which see National Standards Organisation; §10 applies to image processing. The standard has been implemented in XML by the Library of Congress as the MIX schema, which is in turn often used as an extension of the METS standard for technical and administrative metadata. See *MIX* and *METS*. I thank Elena Pierazzo for drawing these to my attention.

not something that even the most computer-literate scholars in the humanities would wish or be able to do. To overcome these three substantial practical hurdles, the prototype system described above automatically logs all of the steps which have been taken, describing those steps in terms of standard and well-documented algorithms, and with all the values of all the settings for each step clearly expressed. This information can then be displayed for the user's reference or saved to disk in a format that conforms to the recognised standard.

This requirement for documentation and reproducibility has one major disadvantage when applied to image-enhancement in that it limits the tools which are available for use. For example, Craig-McFeely has noted that one cannot blindly apply a single process to an entire document but that any successful enhancement requires a great deal of detailed human interaction.[24] Her methods include the painstaking use of particular tools in Adobe Photoshop to work on small and very precise areas of the image at a time. Her methods are extremely effective and should certainly not be dismissed. However, the application of different tools many different times to small areas of the image means that it is almost impossible to record her interventions with sufficient detail and accuracy for them to be reproduced. This leads to something of a conundrum about the extent to which demonstrably valuable techniques should be eschewed in favour of the more theoretical demand for precise documentation. Perhaps improvements in image-processing will require less and less human intervention of this sort and will therefore allow proper documentation to accompany good results. Whatever the case, the ideal would be to use only techniques which are purely statistical, that is, where the same sequence of processes is applied equally to the whole image rather than the closely localised use of different tools and processes in different places. These "whole-image" techniques may very well require careful and extensive human intervention, and we may still be some way from devising techniques which are sufficiently effective, but the final result, in an ideal world, must surely be one that can be recorded and reproduced.

[24] Craig-McFeely §51; compare the techniques listed by Craig-McFeely and Lock 38–50, almost all of which are labour-intensive and apply to only parts of an image at a time which must be selected by the user and which cannot be precisely reproduced.

This problem of accountability slides into what Julia Craig-McFeely calls "ethical enhancement".[25] The question here is at what point "enhancement" becomes "interpretation", and how much of the latter should be permitted. This can be illustrated by reference to Figure 9. This figure shows two small portions of what seems to be perfectly legitimate twelfth-century script, and it probably comes as no surprise to learn that both derive from a genuine, original cartulary which was produced at Bath. However, only one is a photograph of the manuscript; the other is fabricated by using the GIMP to copy letters from the document and to piece them together to form a new word which is almost entirely indistinguishable from the original. This is in essence a more sophisticated version of the old ransom notes which were featured in movies, and formed by pasting together letters cut from newspapers. I would not suggest that researchers deliberately manipulate photographs to produce entirely new readings such as this. However, some procedures for image-enhancement do use much the same method to restore readings which the researchers are certain were there before the manuscript was damaged. This temptation is understandable, particularly as it is not unusual for an enhanced image to be perfectly legible on a computer screen but for this legibility to be lost utterly when the image is printed for publication. Rather than yielding to this temptation, however helpful it may seem as a means of recording one's results, it is preferable to record the exact process of enhancement, as discussed above, and ideally to provide both original and enhanced images.

If any conclusion can be drawn from this then perhaps it is that the enhancement of digital images is useful for recovering Anglo-Saxon erasures and other forms of lost readings. The tools and techniques outlined here serve a valid purpose, and they should not be dismissed wholesale because of the problems they create when misused. It is important for these problems to be properly understood, both by those carrying out the enhancement and by those using and judging the results of such techniques. And with these techniques, tools, and images all readily available, it is now possible at least in principle for all scholars of manuscripts to enhance their own images and recover their own lost readings in one further step towards greater accessibility to manuscripts and the evidence they contain.[26]

[25] Craig-McFeely §62; compare also Craig-McFeely and Lock 35–36 and 53–54.

[26] I wish to thank the Cambridge Newton Trust and the Leverhulme Trust for their financial support, without which this research would not have been possible.

WORKS CITED

Budny, Mildred. *Insular, Anglo-Saxon, and Early Anglo-Norman Manuscript Art at Corpus Christi College, Cambridge: An Illustrated Catalogue*. 2 vols. Kalamazoo, MI: Medieval Institute Publications, 1997.

Craig-McFeely, Julia. "Digital Image Archive of Medieval Music: The evolution of a digital resource." *Digital Medievalist* 3 (2008): 90 paras. 29 May 2008 <http://www.digitalmedievalist.org/journal/3/mcfeely/>.

Craig-McFeely, Julia, and Alan Lock. *Digital Image Archive of Medieval Music: Digital Restoration Workbook*. Oxford: Oxford Select Specialist Catalogue Publications, 2006. 5 June 2008 <http://www.methodsnetwork.ac.uk/redist/pdf/workbook1.pdf>.

Easton, Roger L., Jnr. *Text Recovery from the Archimedes Palimpsest: An Exercise in Digital Image Processing*. 25 July 2001. Rochester Institute of Technology. 29 May 2008 <http://www.cis.rit.edu/people/faculty/easton/k-12/>.

Easton, R. L., Jr, K. T. Knox, and W. A. Christens-Barry. "Multi-spectral imaging of the Archimedes palimpsest." *Proceedings of the Applied Imagery Pattern Recognition Workshop (IEEE-AIPR'03)* 32 (2003). 111–16.

Easton, Roger L., Keith T. Knox. "Digital Restoration of Erased and Damaged Manuscripts." *Proceedings of the 39th Annual Convention of the Association of Jewish Libraries*. Compiled by Elana Gensler and Joan Biella. New York: Association of Jewish Libraries, 2004. 29 May 2008 <http://www.jewishlibraries.org/ajlweb/publications/proceedings/proceedings2004.htm>.

GIMP: GNU Image Manipulation Program. 2008. Michael Natterer and others. 29 May 2008 <http://www.gimp.org>.

Gneuss, Helmut. *Handlist of Anglo-Saxon Manuscripts: A List of Manuscripts and Manuscript Fragments Written or Owned in England up to 1100*. Tempe: Arizona Center for Medieval and Renaissance Studies, 2001.

"GNU General Public Licence." *GNU Operating System.* 29 June 2007. Free Software Foundation. 29 May 2008 <http://www.gnu.org/licenses/gpl.html>.

"Java Advanced Imaging (JAI) API." *Sun Development Network*. 2008. Sun Microsystems. 29 May 2008 <http://java.sun.com/javase/technologies/desktop/media/jai/>.

Ker, Neil R. *Catalogue of Manuscripts Containing Anglo-Saxon.* Oxford: Clarendon, 1957. Revised impression incorporating Supplement, 1990.

Keynes, Simon, and others. "Classified List of Anglo-Saxon Charters on Single Sheets." *Kemble: Anglo-Saxon Charters.* 1 June 2008 <http://www.trin.cam.ac.uk/kemble/singlesheets/ss-index.html>.

Lowe, Elias Avery. "Codices Rescripti: A List of the Oldest Latin Palimpsests with Stray Observations on their Origin." *Mélanges Eugène Tisserant.* Vol. 5. Vatican: Biblioteca Apostolica Vaticana, 1964. 67–113. Reprinted in *Palaeographical Papers*, 1907–1965. Ed. Ludwig Bieler. Vol. 2. Oxford: Clarendon, 1972. 480–519.

——. *Codices Latini Antiquiores: A Palaeographical Guide to Latin Manuscripts Prior to the Ninth Century.* 11 vols. plus supplement. Oxford: Clarendon, 1934–71.

McGurk, Patrick. *An Eleventh-Century Anglo-Saxon Illustrated Miscellany: British Library Cotton Tiberius B. V Part I Together with Leaves from British Library Cotton Nero D. II.* Early English Manuscripts in Facsimile 21. Copenhagen: Rosenkilde and Bagger, 1993.

National Standards Organisation. *Data Dictionary – Technical Metadata for Digital Still Images.* Maryland: National Standards Organisation, 2006. 29 May 2008 <http://www.niso.org/kst/reports/standards/>. Keyword: Z39.87.

METS: Metadata Encoding and Transmission Standard Official Website. 12 May 2008. Library of Congress. 23 May 2008 <http://www.loc.gov/standards/mets/>.

MIX: NISO Metadata for Images in XML Schema. 13 May 2008. Library of Congress. 23 May 2008 <http://www.loc.gov/standards/mix/>.

Nave, Carl. *Hyperphysics.* 2005. Georgia State University. 21 August 2008. <http://hyperphysics.phy-astr.gsu.edu>

Netz, Reviel, and William Noel. *The Archimedes Codex: Revealing the Secrets of the World's Greatest Palimpsest.* London: Weidenfeld & Nicolson, 2007.

Parker on the Web. Stanford University. 1 June 2008 <http://parkerweb.stanford.edu>.

Prescott, Andrew. "'Their Present Miserable State of Cremation': the Restoration of the Cotton Library." *Sir Robert Cotton as Collector: Essays on an Early Stuart Courtier and His Legacy*. Ed. C. J. Wright. London: British Library, 1997. 391–454.

Rinascimento Virtuale - Digitale Palimpsestforschung - Rediscovering Written Records of a Hidden European Cultural Heritage. 29 May 2008 <http://www1.uni-hamburg.de/RV/>.

Salerno, Emanuele, Anna Tonazzini and Luigi Bedini. "Digital Image Analysis to Enhance Underwritten Text in the Archimedes Palimpsest." *International Journal on Document Analysis and Recognition* 9 (2007). 79–87.

Schipper, William. "Digitizing (Nearly) Unreadable Fragments of Cyprian's 'Epistolary'". *The Book Unbound: Editing and Reading Medieval Manuscripts and Texts*. Ed. Siân Echard and Stephen Partridge. Toronto: U of Toronto P, 2004. 159–68.

Stokes, Peter A. *English Vernacular Script, ca. 990 – ca. 1035*. Diss. Cambridge University, 2006.

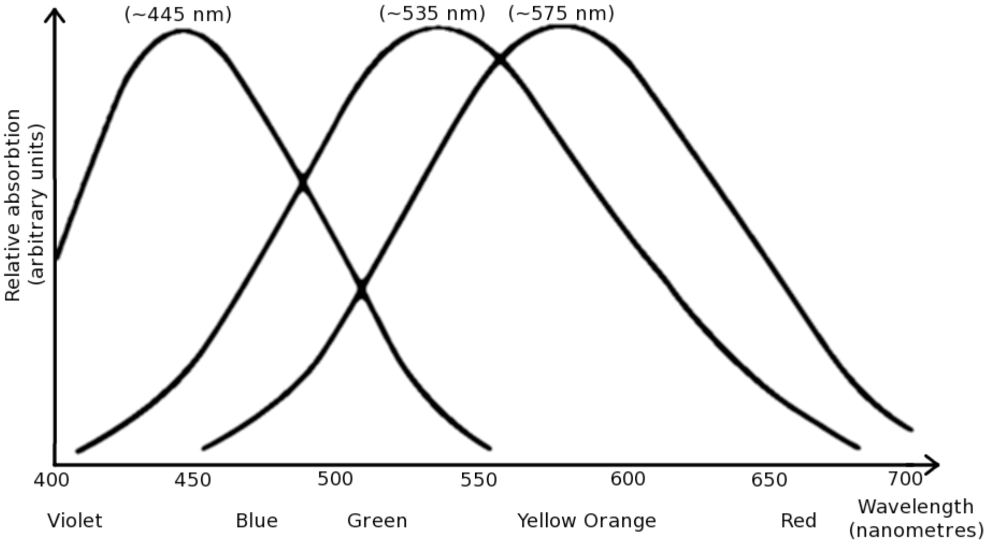


Figure 1: Intensity curves
Based on Nave, "The Color Sensitive Cones".

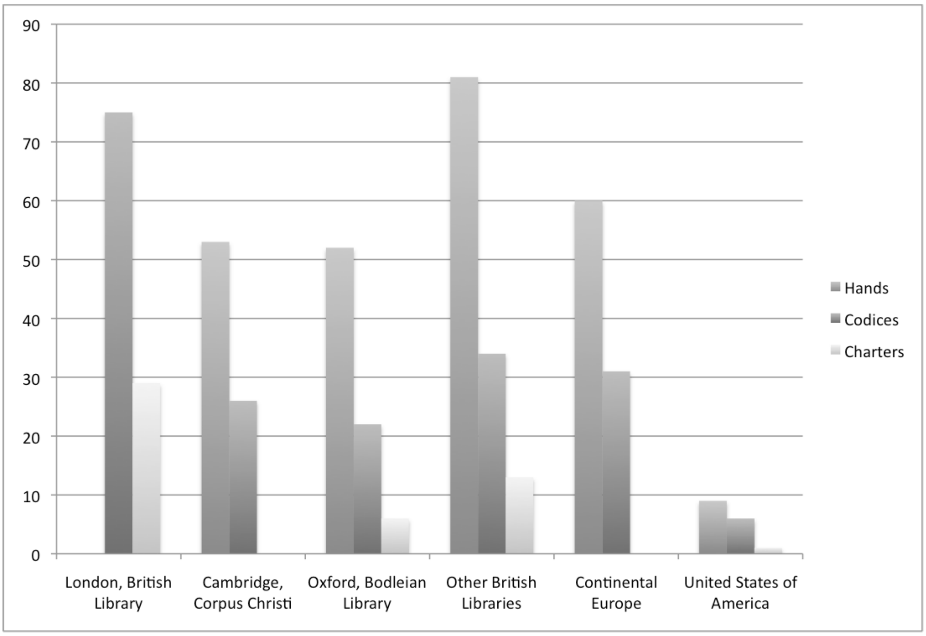
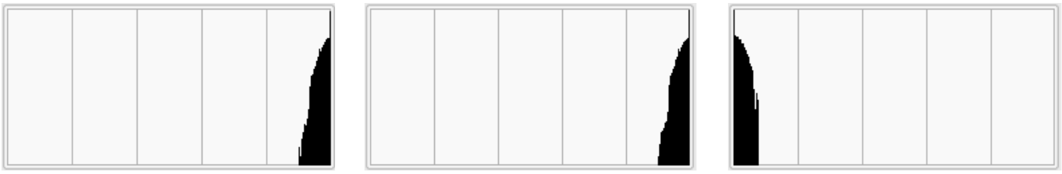
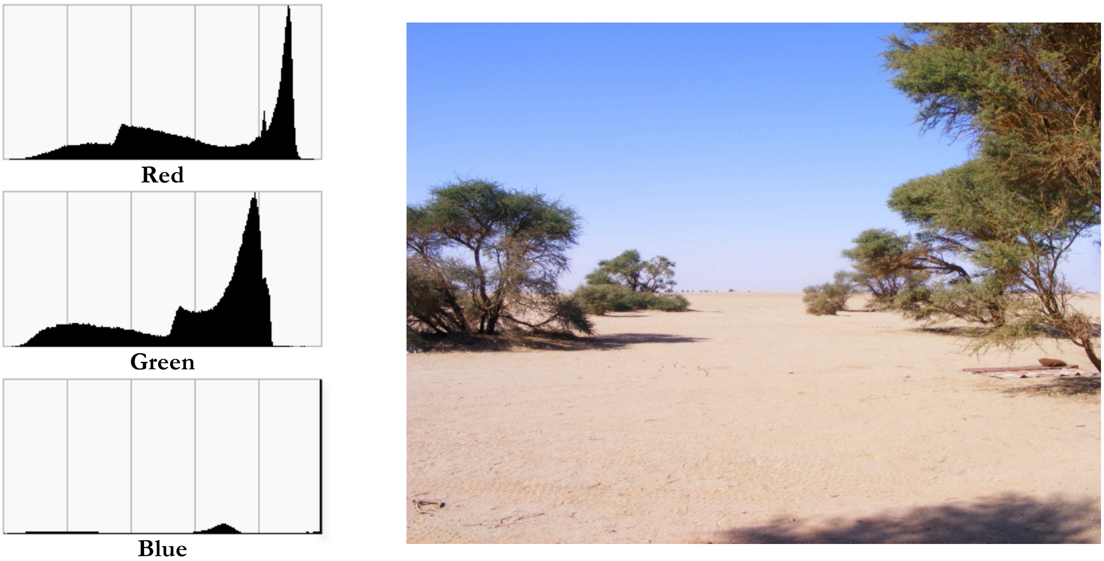


Figure 2: Example of a histogram



Red **Green** **Blue**

Figure 3: Histograms for a yellow square
Histograms are from screenshots of the GIMP



Red

Green

Blue

Figure 4: Histograms for a desert scene (note very narrow peak at extreme right of blue histogram)
Photograph by the author; histograms are from screenshots of the GIMP

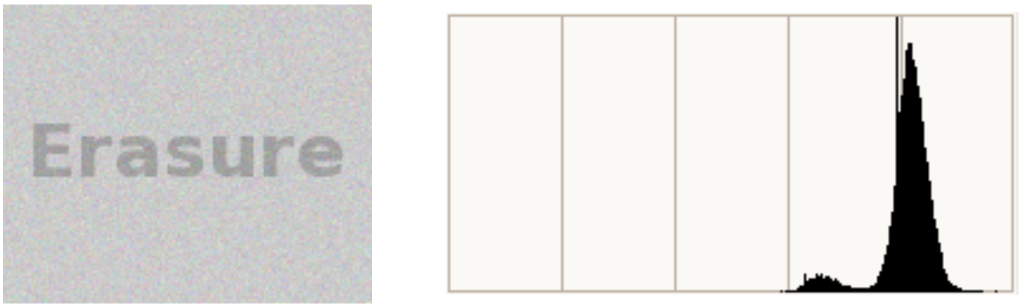


Figure 5: Model "erasure" and accompanying histogram (in greyscale)
Histogram is from a screenshot of the GIMP

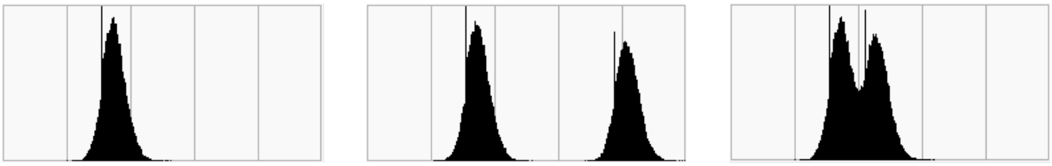
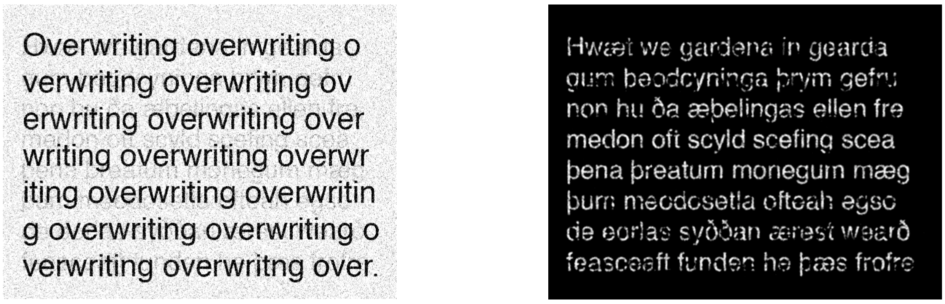


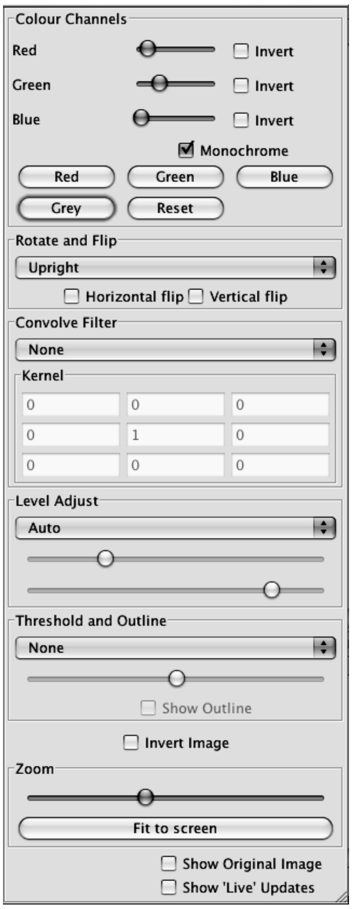
Figure 6: Three types of histogram
Histograms are from screenshots of the GIMP



Overwriting overwriting o
verwriting overwriting ov
erwriting overwriting over
writing overwriting overwr
iting overwriting overwritin
g overwriting overwriting o
verwriting overwritng over.

Hwæt we gardena in gearda
gum þeodcyninga þrym gefru
non hu ða æþelingas ellen fre
medon oft scyld scefing scea
þena þreatum monegum mæg
þum meodosetla ofteah egso
de eorlas syððan ærest wearð
feasceaft funden he þæs frofre

Figure 7: Results of channel manipulation. The left is original, the right enhanced.



Colour Channels

Red □ Invert

Green □ Invert

Blue □ Invert

☑ Monochrome

Rotate and Flip

Upright

□ Horizontal flip □ Vertical flip

Convolve Filter

None

Kernel

0	0	0
0	1	0
0	0	0

Level Adjust

Auto

Threshold and Outline

None

□ Show Outline

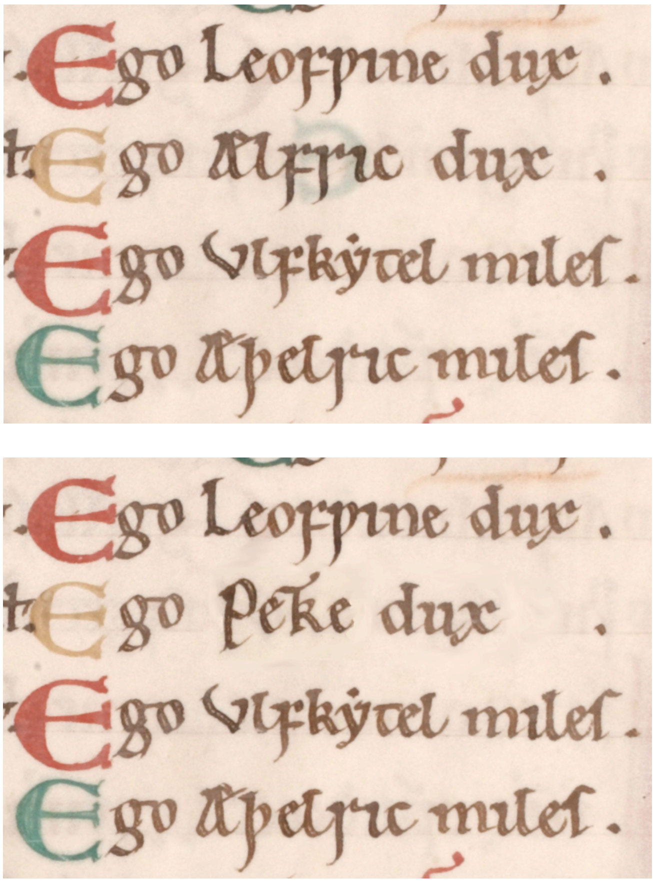
□ Invert Image

Zoom

□ Show Original Image

□ Show 'Live' Updates

Figure 8: ImageViewer Control Panel (on Mac OS X)



Ego Leofƿine dux.
Ego Ælfric dux.
Ego Vlfkẏtel mileſ.
Ego Æþelric mileſ.

Ego Leofƿine dux.
Ego Ƿeke dux.
Ego Vlfkẏtel mileſ.
Ego Æþelric mileſ.

Figure 9: An improperly "enhanced" image. The upper one is original, the lower altered.
CCCC 111, p. 88 (detail), by permission of the Master and Fellows of Corpus Christi College, Cambridge.