

Data Embedding Using Phase Dispersion

Chris Honsinger
Eastman Kodak Company
Rochester, New York, USA

Abstract

We present a method of data embedding based on the convolution of message data with a random phase carrier. We review the theory of the method and show how the technique can be used to hide both pictorial and non-pictorial data. Design considerations for the carrier are also presented. Applications of the technique to robust and fragile authentication are also briefly described. We also present an overview of our rotation and scale correction algorithm and synchronization procedure. Finally, we present the algorithms' Stirmark benchmark result.

Introduction

We adopt the notation that an original image is represented as the two dimensional array, $I(x,y)$, the embedded image, $I'(x,y)$, and we define a carrier as $C(x,y)$. The message that is embedded, $M(x,y)$, in its most general form is an image. The message can represent an icon, for example, a trademark, or may represent the bits in a binary message. In the latter case the on and off states of the bits are represented as plus and minus ones, or positive and negative delta functions which are placed in predefined and unique locations across the message image.

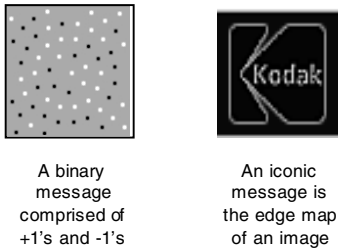


Figure 1. Examples of binary and iconic messages

With these definitions the embedding equation[1] is:

$$I'(x,y) = \alpha(M(x,y) * C(x,y)) + I(x,y), \tag{1}$$

where the symbol, $*$, represents circular convolution and α is an arbitrary constant chosen to make the embedded energy simultaneously invisible and robust to common processing. Recall from Fourier theory that spatial convolution in the frequency domain is the same as adding phase while multiplying magnitudes. Therefore, the effect of convolving the message with a carrier is to distribute the message energy in accordance with the phase of the carrier and to modulate the amplitude spectrum of the message with the amplitude spectrum of the carrier. If the message were a single delta function and the carrier of random phase and of uniform Fourier magnitude, the effect of convolving with the carrier would be to distribute the delta function over space. The effect of convolving a message with a random phase carrier is to spatially disperse the message energy.

The basic extraction process is to correlate with the same carrier used to embed the image:

$$\begin{aligned} I'(x,y) \otimes C(x,y) &= \alpha(M(x,y) * C(x,y)) \otimes C(x,y) \\ &+ I(x,y) \otimes C(x,y), \end{aligned} \tag{2}$$

where the symbol, $\otimes$, represents circular correlation. Correlation is similar to convolution in that Fourier magnitudes also multiply. In correlation, however, phase subtracts. Therefore, the phase of the carrier subtracts on correlation of the embedded image with the carrier leaving the message. Indeed, if we assume that the carrier is designed to have uniform Fourier amplitude, then $C(x,y) \otimes C(x,y) = \delta(x,y)$, and the process of correlation of the carrier on the embedded image Eq. 2, can be reduced to:

$$I'(x,y) \otimes C(x,y) = \alpha M(x,y) + noise \tag{3}$$

That is, the process of correlation of the embedded image with the carrier reproduces the message image plus noise due to the cross correlation of the image with the carrier.

Experimentally, we have found that tiling the dispersed message on the original image improves the robustness of the algorithm. With rare exception, we tile a single 128x128 dispersed message over the entire image. Upon extraction, each 128x128 region is aligned and summed to produce the final message. For imaging applications with severe quality loss, such as small images printed using ink-jet printers on plain paper, a weighting factor that depends on the estimated signal to noise ratio is calculated and applied to each extracted message element before summation.

Carrier Considerations

If we denote the extracted message as $M'(x,y)$, we can rewrite the equations for extracting the message (Eq. 2 and Eq. 3), above as:

$$M'(x,y) = \alpha M(x,y) * (C(x,y) \otimes C(x,y)) + noise \quad (4)$$

The above equation suggests that the resolution of the extracted message is fundamentally limited by the autocorrelation function of the carrier, $C(x,y) \otimes C(x,y)$. Any broadening of $C(x,y) \otimes C(x,y)$ from a delta function will blur the extracted message when compared to the original message. Another way to view the effect of the carrier on the extracted message is to consider $C(x,y) \otimes C(x,y)$ as a point spread function, since convolution of the original message with $C(x,y) \otimes C(x,y)$ largely determines the extracted message.

The design of the carrier should consider both the visual detectability of the embedded signal and the expected signal quality at the extraction step. There is clearly a design tradeoff between achieving optimum extracted signal quality and embedded signal invisibility.

A carrier designed for optimal extracted signal quality will possess increasing amplitude with increasing spatial frequency. This may be derived from the well-known characteristic of typical images that the Fourier amplitude spectrum falls as the inverse of spatial frequency. At low spatial frequencies, where typical images have their highest energy and influence on the extracted image, our carrier uses this result. In particular, the mean or DC frequency amplitude of our carrier is always zero. As spatial frequency is increased, the carrier amplitude envelope smoothly increases with increasing spatial frequency until about 1/16 to 1/5 Nyquist.

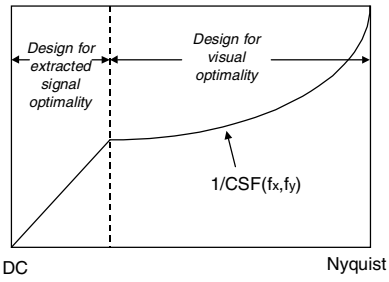


Figure 2. The carrier is designed for extracted signal quality for low spatial frequencies and visual optimality for high spatial frequencies

As shown in Figure 2, for frequencies greater than this, the carrier envelope is derived from a Contrast Sensitivity Function[2] (CSF). The CSF provides a measure of the sensitivity of the average observer to changes in contrast at a given spatial frequency. The reciprocal of the CSF can be used to prescribe the amount of amplitude needed for the embedded signal to be detectable by an average viewer. Many modern CSF models facilitate for observer viewing distance, background noise, device dot density, color component wavelength and other factors.

Use of these CSF parameters can be an advantage when optimizing an embedding algorithm for a specific application. One particularly useful way of sizing the embedding algorithm for a specific system is to define the quality of the embedded signal in terms of the viewing distance at which the embedded signal can be visually detected. Once this is defined, an optimized carrier can be immediately derived and tested. Figure 3 shows the wide variation of embedded energy possible due to observer viewing distance in the frequency domain.

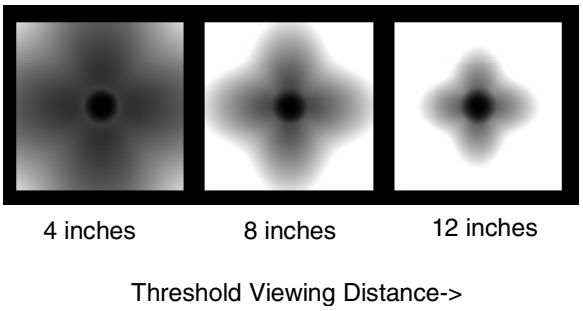


Figure 3. Variation of Fourier magnitude of the carrier as a function of viewing distance (Device dpi=300)

For a binary message, the impact of this carrier envelope is to produce a very small sidelobe around each delta function. It may be argued that the sidelobes rob the algorithm of bandwidth. However, we have found that the destructive processes of compression, error diffusion, printing and scanning have a far greater influence on the bandwidth of the algorithm. In a binary message, these destructive processes are the limiting factor of the bit density and can be thought of as defining the minimum separation distance between the delta functions. So long as the sidelobes are confined within half of the minimum bit separation distance, sidelobe interference may be considered minimal.

Message Types

An important aspect of our technique is that it can be used to embed either iconic or binary data types. Examples of iconic data types are trademarks, corporate logos or other arbitrary images. Performance generally decreases as the message energy increases so edge maps of the icons are used.

Obvious examples of binary data types are 32 bit representations of URL's or copyright notices. A binary data type is also represented as an image with each bit represented as a positive or negative spike. Typically, we use between 32 and 256 bits per image.

Iconic Data Types

Trademark

Figure 4 shows an iconic message before embedding and after extraction. The advantage of embedding an iconic message over an abstract type is that the iconic message can be interpreted directly. The human visual system can be used to interpret through what may be an otherwise prohibitively large amount of noise.

Figure 4. Original and extracted iconic image. Each image is 128 pixels x 128 lines

Authentication

One compelling application of the technique is the idea of embedding an image's own edge map in the image. In this case we abandon the idea of tiling a dispersed message over the image and use the full image extent as our message and carrier extent. Since the carrier is of random phase, the edge map is distributed randomly over the image. Each pixel of the image contains a contribution from every pixel of the edge map. Attempting to deceive an audience by altering a small region of the image by replacing with another image is futile because the original edge map is extractable.

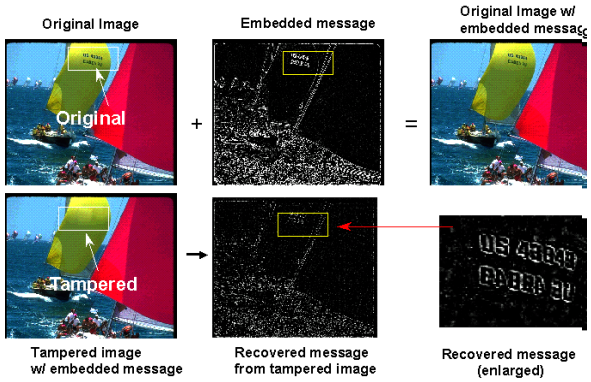


Figure 5. Authentication using an image's edge map

For example, in top row of Figure 5 we show the steps of embedding the image of a sailboat with its edge map. In lower left corner, we show the sailboat with its sail altered so that the identification symbols are no longer visible. To obtain this image, we completely removed the region containing the symbols and replaced the region with an image of the same color and approximate texture. The replaced region contained no embedded data. In the center of the bottom row of Figure 5, we show the extracted edge map, and in the lower right corner we show that the edge map data containing the sail identifier is, indeed, extractable from the altered image.

Unlike traditional forms of authentication, where no changes to the original image are allowed, this technique has promise in its robustness to printing and scanning.

Binary Data Types

The advantage of binary data types is that a human does not need to be involved in the interpretation of the extracted message. There are numerous applications of using binary data ranging from the interruption of a scanner due to the detection of a digital copyright notice to transmission of metadata provided by a digital camera.

Authentication

In traditional forms of image authentication, an encrypted hash of the image is sent as an appendage to the image. At the receiver, the hash is decrypted and compared to a recalculated hash of the image. If the hashes match, the image is declared authentic. Otherwise, the image is considered suspicious.

We have demonstrated a technique that allows the encrypted hash to be losslessly embedded of the image. That is, no appendage is required. We assume that access to the original image is available such as within a digital camera. Furthermore, we restrict the values of the original image to the range (1-254). For each 128x128 region of the image a 160-bit hash is calculated and encrypted. This message is convolved with the carrier and thresholded to produce a 128x128 dispersed message comprised entirely of zeros and plus ones and minus ones (See Eq. 5). The thresholded dispersed message is added to the image to produce an image with a dynamic range of (0-255).

$$M_{thresh} = \begin{cases} 0, & |M| < t \\ sign(M), & |M| \le t \end{cases} \qquad (5)$$

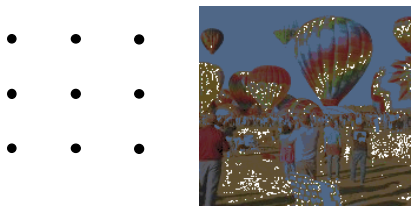
To authenticate the image, the image is correlated with the carrier and the encrypted hash is extracted. The encrypted hash is then used to regenerate the dispersed message. The dispersed message again thresholded to regenerate what was added to the image, but now the thresholded dispersed message is subtracted. At this point, the original data is available and the hash is recalculated, and is compared to the decrypted extracted hash. If they are equal, the image is authentic, if they are not equal, the image is not authentic.

We have successfully demonstrated this technique for encrypted hash lengths of up to 225 unique bits for each (128x128 region).

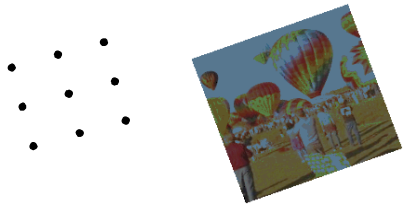
Rotation/Scale

Correcting for rotation and scaling is a fundamental element of all robust data embedding techniques. Our rotation scale correction technique relies on autocorrelation the embedded image[3]. For example, upon autocorrelation of an embedded image that has not been rotated or scaled, we would expect to see correlation peaks spaced horizontally and vertically at intervals of 128 pixels and 128 lines. At the zero offset correlation point, there is a very high peak due to the image correlating with itself.

Now, if the embedded image is scaled, the peaks must scale proportionately. Similarly, if the embedded image is rotated, the peaks must rotate by the same amount. Therefore, the rotation and scale of an image can be deduced by locating the autocorrelation peaks. (See Figure 6.) Detection of the actual rotation angle θ is limited to angles in the range (-45° ,+45°]. However, the actual rotation angle will be a member of the set $\theta_{actual} = \theta_{calculated} \pm n90°$, where n is an integer. Because we test for the possibility that the image has been flipped in multiple directions in the message extraction process, this ambiguity is not a limitation.



Autocorrelation of an embedded image that has not been rotated or scaled



Autocorrelation of an embedded image that has been rotated and scaled

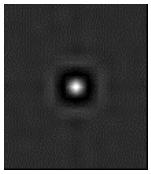
Figure 6. The concept of autocorrelating an embedded image to derive scale and rotation parameters

The effect of the autocorrelation properties of the original image can be significant. Without ancillary processing, high amplitude low frequency interference in the autocorrelation image can make the process of detecting peaks difficult. To minimize this problem, we perform localized first order and second order moment normalization on the embedded image before the autocorrelation. This process consists of replacing each pixel in the image with a new pixel value, v_{new} :

$$v_{new} = \frac{\sigma_{desired}}{\sigma_{old}}(v_{old} - m_{old}) \qquad (6)$$

where v_{old}, is the original pixel value, m_{old} , is the local mean of the image, $\sigma_{desired}$ is the desired standard deviation, which is generally set to the expected embedded signal standard deviation and σ_{old} is the local standard deviation. Because this operation is over a small area, typically over a (3x3) or (5x5) region, its effect in removing the high amplitude, low frequency coherent noise is quite substantial. For the limiting case when $\sigma_{old} \rightarrow 0$, we simply equate v_{new} to a value taken from a random noise generator having a standard deviation $\sigma_{desired}$.

The next piece of ancillary processing we perform is to shape the autocorrelation peaks. This is done during the FFT operation used in the autocorrelation processing. We have found that a function that increases linearly with spatial frequency in the Fourier magnitude domain to be quite satisfactory. This function is consistent with a Wiener filter designed to maximize the semblance of the correlation peaks to delta functions under the assumption that the image Fourier amplitude spectrum exhibits an asymptotic "1/(spatial frequency)" falloff. Following these processing steps produces peaks (see Figure 7) that need little further processing.



Enlarged autocorrelation peak

Figure 7. An actual autocorrelation peak (enlarged) demonstrating high signal to noise ratio

Importantly, because autocorrelating the embedded image requires no extra calibration signal, it does not tax the information capacity of the embedding system. In addition, this technique can be applied to any embedding technique with redundant embedded signals and may implemented on a more local level to confront low order geometric warps.

Synchronization

The ability to recover from cropping is an essential component of a data embedding algorithm. Using our algorithm, if we were to extract from an arbitrarily located 128x128 region of an embedded image, the extracted message would probably appear to be circularly shifted due to the unlikely chance that the extraction occurred along the original message boundary.

Indeed, if the origin of the 128x128 extracted region was a distance, $(\Delta x, \Delta y)$, from its nearest "original" origin, then the extracted message, $M'(x,y)$ can be written as:

$$M'(x,y) = M(x,y) * \delta(x - \Delta x, y - \Delta y) \qquad (7)$$

where we have assumed that the convolution is circular, that the carrier autocorrelated to a delta function and that the image contributes no noise.

On the surface, this circular shift ambiguity is a severe limitation on data capacity because it imposes the constraint that the message structure must be invariant to circular shifts. However, we have found a way around this by placing the bits in the message in a special manner. First, we require the use of a *message template*, that is, a prescription of where to place the bits in a message image. The message template is derived by placing positive delta functions on a blank 128x128 image such that each delta function is located a minimum distance away from all others and such that the autocorrelation of the message template yields as close as possible, a delta function. That is, we place the bits such the message template autocorrelation sidelobes are of minimal amplitude.

Now, correlation of the extracted region with a zero mean carrier guarantees that the extracted circularly shifted message $M'(x,y)$ is also zero mean. If we call the message template, $T(x,y)$, then the absolute value of the the extracted template must be practically equivalent to a circularly shifted message template. That is,

$$\left|M'(x,y)\right| = T(x,y) * \delta(x-\Delta x, y-\Delta y) \tag{8}$$

This implies, due to the autocorrelation property of the message template, that the shift from the origin of the message can be derived by correlating $|M'(x,y)|$ with $T(x,y)$, since:

$$\left|M'(x,y)\right| \otimes T(x,y) = \delta(x-\Delta x, y-\Delta y) \tag{9}$$

That is, the result of the correlation will be a 128x128 image, whose highest peak will be located at the desired shift distance, $(\Delta x, \Delta y)$.

Stirmark Results

One way to assess the effectiveness of a data embedding algorithm is by using the StirMark[4,5] algorithm benchmark. Briefly, the Stirmark algorithm exposes an embedding algorithm to many types of adverse processing such as rotation/scale, compression, shearing, warping and other processes that a robust data embedding technique would need to handle. The algorithm described in this paper scores a 0.81 using StirMark 3.0 with a 64 bit binary payload and a uniform amplitude carrier adjusted to a PSNR of 38dB. More information may be obtained on this benchmark by visiting: *http://www.cl.cam.ac.uk/~fapp2/steganography*.

Summary

We have shown that data embedding using phase dispersion may be used for fragile and robust data hiding applications. Because the technique does not rely on a calibration signal for rotation and scale correction, the information capacity may be increased over competing technologies.

References

1. S. J. Daly, J. R. Squilla, M. Denber, C. W. Honsinger J. Hamilton, "Method for embedding digital information in an image", U.S. Patent 5,859,920, 1999.
2. S. J. Daly, "Method and apparatus for hiding one image or pattern within another", U.S. Patent 5,905,819, 1999.
3. C. W. Honsinger, S. J. Daly, "Method for detecting rotation and magnification in images", U.S. Patent 5,835,639, 1998.
4. Fabien A. P. Petitcolas, Ross J. Anderson, Markus G. Kuhn. Attacks on copyright marking systems, in David Aucsmith (Ed), Information Hiding, Second International Workshop, IH'98, Portland, Oregon, U.S.A., April 15-17, 1998, Proceedings, LNCS 1525, Springer-Verlag, ISBN 3-540-65386-4, pp. 219-239.
5. Fabien A. P. Petitcolas and Ross J. Anderson, Evaluation of copyright marking systems. In proceedings of IEEE Multimedia Systems (ICMCS'99), vol. 1, pp. 574--579, 7--11 June 1999, Florence, Italy.

Biography

Chris Honsinger received his B.S. and M.S. degree in Physics from the Ohio University in 1983. From 1983 to 1985 he worked as a Geophysicist for Amoco Production Company in Houston Texas. From 1985 to 1995 he worked on government imaging systems for Kodak's Commercial and Government Systems Division. Since 1995 he has been with the Kodak Research Laboratories Image Compression and Security Group.