Fast Mobile Stegatone Detection using the Frequency Domain

Robert Ulichney, Hewlett-Packard Co., Andover, MA, USA
Stephen Pollard, Hewlett-Packard Co., Bristol, UK
Matthew Gaubatz, Hewlett-Packard Co., Seattle, WA, USA

Abstract

Data-bearing halftone images are an attractive high-capacity alternative to barcodes. We show that smartphone close-focus video capture can recover the associated data, and a frequency-domain-based algorithm enables a robust means of detecting the presence as well as determining the location, scale, and orientation of data-bearing designs. Android and iOS mobile implementations achieve this at near video rates. Using these affine transform parameters as a starting point, a method for full perspective correction for precise alignment is also developed. This functionality is fast, accurate, and achieved without the benefit of fiducial marks.

Introduction

Barcodes are useful for labeling but can degrade the aesthetics of documents. Watermarks can offer near-invisible impact on images, but the data capacity is limited. Using image recognition to associate an index with each image requires no change to a printed image but data recovery will only work for a small set of index values which is even more limiting on data density. While not covert like watermarks or image matching, embedding data in clustered dot halftones by shifting clusters, as in stegatones [1] or by dot orientation [2], yields a high data density alternative to barcodes. Mobile barcode reading apps achieve a fast reaction time because barcode fiducials (such as the large square targets on QR codes) are easy to find automatically. In this paper we describe a solution for detecting stegatones at nearly the video frame rate on smartphones, and for accurately determining scale, rotation, and perspective distortion.

Previous stegatone recovery solutions assumed that the print and capture resolution of the stegatone were known to the system and thus avoided the need to recover the scale at run time. Part of the issue is that solutions for detecting image objects have been tested on scanned imagery, which is considerably easier to interpret. It has been shown that individual halftone dots can be detected [3], and in principle, this type of approach could be combined with subsequent processing to identify the marking period, which again in principle, could be used as part of the proposed scheme. It relies on the Hough transform, however, which in some cases cannot be used simply due to computational complexity. Other approaches involving dot size estimates have been proposed for clustered-dot halftones, but for a compression application [4]. An approach that indirectly estimates the cell size via descreening process has been proposed [5], but again, is based on techniques that may not be robust or simply fast enough for an efficient cell-phone-based approach. A method for detecting halftone frequency has been disclosed for the purpose of optimizing a print production pipeline. This mechanism, however, uses original (digital) imagery as opposed to data from mobile devices, and is based on the computation of an autocorrelation function.

One type of solution that in principle would be applicable to the problem is the Viola-Jones object detection framework [6]; while this approach excels at finding macroscopic objects it is far too coarse for locating elements of halftone structure. Similarly, methods such as SIFT-based [7] or SURF-based [8] feature detectors, or any one of their variants, and are too slow and cumbersome for this application where the number of near identical features can grow very large at finer resolutions. Using the discrete Fourier transform (DFT) to measure the fundamental frequency of halftone patterns, on the other hand, is a well-known technique, and is an important tool for assessing the interaction of overlaying color screens.

In this paper we show that we can use the DFT to solve the detection problem and that current smartphones can perform the computation in real time. Our measurements of current Android and iOS smartphones indicated a closest focus limit of 8 cm, with a plot of video capture resolution plotted in Figure 1. To increase the range of distances at which we can capture and recover stegatones, designs are printed at 400 dpi (100 halftone cells per inch). The plot shows a capture resolution of 400 dpi at a distance of about 17cm. This affords a workable capture distance range of 9 cm for capturing and resolving the single pixel shifts needed for data recovery.

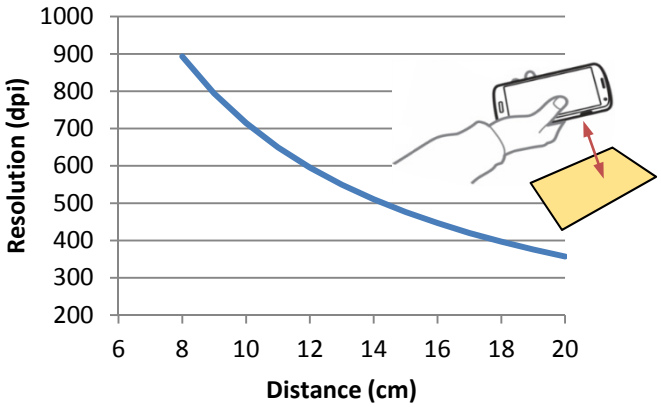

Figure 1. *Close-focus mobile video capture resolution.*

Frequency Domain Peaks

Consider the example mobile capture of a prototype stegatone passport security feature in Figure 2. As with any hand held capture result, the image suffers from rotation and some perspective distortion. The central portion of the DFT of this capture is shown using false color, in Figure 3, where the axes are given in units of cycles/captured-pixel. The disc at the center is a

mask to block low frequency components. Even though the stegatone in Figure 2 covers less than 14% of the captured video frame, the periodic structure of the halftone produces very strong and dominant spikes in the frequency domain which are straightforward to locate automatically.

Figure 2. *Oblique and rotated mobile capture.*

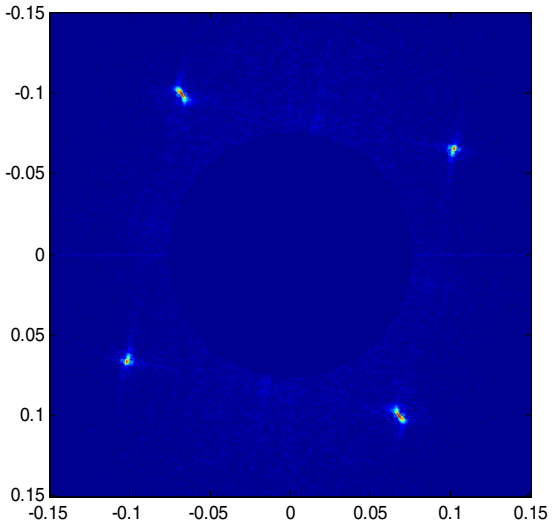

Figure 3. *DFT of image in Figure 2.*

Stegatones use a classical 45-degree screen. Figure 4 depicts an enlarged view of the arrangement of highlight and shadow cells of such a halftone. The cells are square and have a width of q printed pixels. The fundamental spatial period of this screen is r printed pixels. For a typical value of q=4, r is $4\sqrt{2}$.

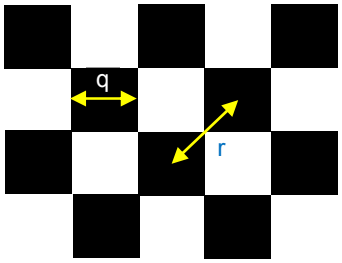

Figure 4. *Checkerboard screen, units of printed pixels.*

A highly stylized example of a captured version of the printed halftone of Figure 4 is illustrated in Figure 5. For simplicity, we model the image of the captured halftone as having horizontal and vertical cell dimensions of h and v (in capture-pixels) and a rotation angle of θ. While not an accurate model of the perspective distortion of the halftone, it is a useful approximation which also allows us to introduce the properties of the DFT.

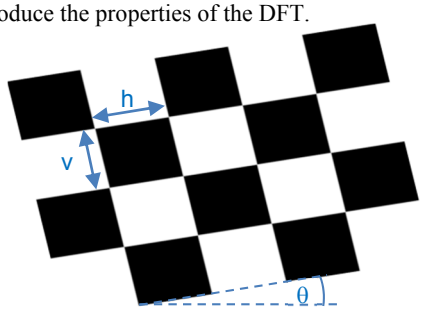

Figure 5. *Stylized captured version of Figure 4.*

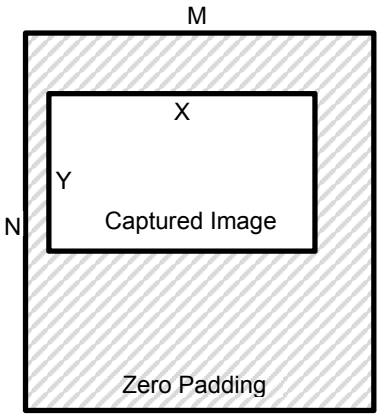

Figure 6. *Padding with zeroes to enhance DFT Resolution.*

When applying the DFT to captured video frames of size X by Y pixels, the region is padded with zeroes, resulting in a frame of size M by N (see Figure 6). (In the simplest case, i.e., without any zero padding, M=X and N=Y.) The M by N DFT amplitude reveals four dominant peaks p_1, p_2, p_3 and p_4. The arrangement of recovered peak locations around the DC coefficient is depicted in Figure 7. Because we are operating on real (as opposed to complex) pixel values, p_3 is a reflected copy of p_1 through the origin, both at a distance of Z_1 from the DC. Likewise p_4 is a reflected copy of p_2 at a distance of Z_2. The distances between p_1 and p_2i, and p_1 and p_4, are given by H and V, respectively. The distances H and V can be normalized to have units of cycles/captured-pixel as follows:

$$H' = \sqrt{\left(\frac{H_x}{M}\right)^2 + \left(\frac{H_y}{N}\right)^2}\ ,\ V' = \sqrt{\left(\frac{V_x}{M}\right)^2 + \left(\frac{V_y}{N}\right)^2}.$$

An estimate of the rotation angle is then

$$\theta = \arctan\left(\frac{H_y/N}{H_x/M}\right).$$

A similar estimate can be obtained from V_x and V_y.
Note that for the case where M=N, these equations simplify to:

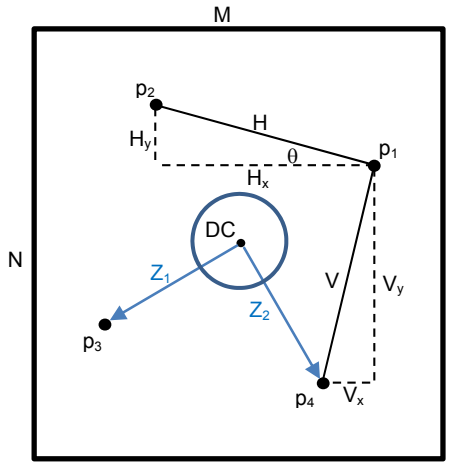

***Figure 7**. DFT peak locations and related measurements.*

$H' = H/N$, $V' = V/N$, and $\theta = \arctan(H_y/H_x)$.

The number of horizontal (h) and vertical (v) capture-pixels per cell are derived as follows:

$$h = {}^{1}/_{H'} \text{ and } v = {}^{1}/_{V'}.$$

The horizontal and vertical scale estimates (S_H, S_V) are given by

$$S_H = {}^{h}/_{q} \text{ and } S_V = {}^{v}/_{q},$$

and thus capture resolutions (C_H, C_V) are estimated

$$C_H = S_H P = {}^{hP}/_{q} \text{ and } C_V = S_V P = {}^{vP}/_{q},$$

where P is the print resolution.

Analysis of the DFT peaks associated with the example from Figure 2 and Figure 3 yields an estimated rotation angle of 11.2°, a stegatone cell size of h=5.79 by v=5.95 capture pixels, and a capture resolution of C_h=578.9 by C_v=595.4 dpi. Examination of the peaks in Figure 3 reveals that the pattern as a whole is skewed, and that rather than exhibiting sharpness the peaks exhibit some elongation. This behavior occurs because the capture axis was not perfectly perpendicular to the target. The resulting perspective distortion caused the image to be skewed and to demonstrate minor spatial variation in scale and resolution, manifested as this peak elongation. These limitations in the model used to describe the transformation of the halftone will be addressed in the next section. Nevertheless, the information already recovered is useful in deciding if the captured image includes a potential stegatone of the required resolution/scale for successful decoding, and if it has not been captured at too oblique a viewing angle (by setting a limit on the ratio of the horizontal and vertical scale parameters).

For best performance, filtering the DFT coefficients by applying a DC mask is an important design consideration. If the mask is too large it can obscure peaks associated with close focused targets; if too small it can leak extraneous DFT energy and confuse the location of the true peaks. Thus, it is useful to characterize the minimum DC-to-peak distance Z that preserves the desired information. The average value of Z is normalized to find Z' in the same manner as H' and V'. Z' is a function of the fundamental spatial period r (in units of printed-pixels/cycle), the print and capture resolutions P and C, that is,

$$Z' = P/(Cr).$$

Using data in Figure 1, we know the capture resolution at closest focus is approximately C=900 dpi. For stegatones printed at P=400 dpi and a cell size of 4 (with r=4√2), the shortest distance Z that preserves the desired structure is 0.078 cycles/captured-pixel. For these values, a reasonable radius of the DC Mask is 0.075, as is used in Figure 3.

Accuracy vs. Speed

There is a tradeoff between the accuracy and speed of computation based on the size of the DFT. Figure 8 exhibits a simple 1D example of how insufficient DFT samples can cause inaccurate estimates of peak location. The blue curve represents the continuous Fourier transform that is sampled by the DFT. The red dots are the samples at a particular resolution, and show how both the location and amplitude of the true peak are missed. One solution is to capture a larger image size to increase DFT resolution. Another alternative is to pad the image with zeroes as was shown in Figure 6; the zero padding in the spatial domain has the effect of smoothing the original DFT and re-sampling with the following interpolation function [9]:

$$f(u,v) = \sin(\pi u)\sin(\pi v)/(MN\sin(\pi u/M)\sin(\pi v/N))$$

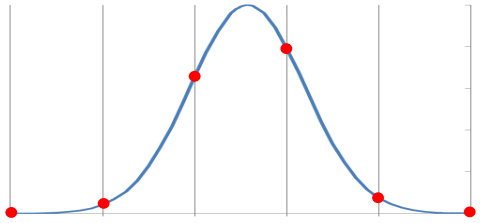

***Figure 8.** Location and amplitude inaccuracy of DFT sampling.*

Alternatively, the peak location can be more accurately interpolated by direct spatial-domain convolution with a local kernel. Note that apart from enabling binary determination of whether or not the stegatone is present in the field of view, the accuracy of the system only has to be good enough to provide a starting point for the alignment system that further corrects for perspective distortion.

Recovering Affine and Planar Projections

We begin this section by extending our model of the imaging process to account for the skew in the locations of the fundamentals. It is shown that we can use knowledge of the halftone image from which the stegatone was constructed to find an estimate of the location of the transformed stegatone in the captured image, and then to recover the associated full planar-perspective distortion to a high degree of accuracy. Consider first the two fundamentals in the top half of the DFT at locations $p_1=(u_1, v_1)$ and $p_2=(u_2,v_2)$ in normalized coordinates. Representing the affine parameters of the transformation that map locations in the DFT of the undistorted halftone to that of the captured image (coordinates with a dash)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

and solving for the transformation of the 2 fundamentals in the DFT we obtain

$$\begin{aligned} a_1 &= (u_1 - u_2)/Q\,, \\ a_2 &= (u_2 + u_1)/Q\,, \\ a_3 &= (v_1 - v_2)/Q\,, \\ a_4 &= (v_2 + v_1)/Q\,. \end{aligned}$$

The parameter $Q = 1/q$ represents the normalized displacement parameter of the undistorted fundamentals in the DFT which are located at $(Q/2\,, Q/2)$ and $(-Q/2\,, Q/2)$ respectively for p_1 and p_2 with respect to the DC. The affine parameters in the image space are related to those in the DFT [10] (transpose of the inverse):

$$A = \frac{1}{a_1 a_4 - a_2 a_3} \begin{bmatrix} a_1 & -a_3 \\ -a_2 & a_4 \end{bmatrix}.$$

Translation, and subsequently planar projection, can be recovered using gradient decent like methods in the image domain. We find it most efficient and robust to find the approximate translation first, using the scale and affine parameters recovered from the DFT. That is, we apply equivalent low pass filters to the central region of the captured image and the reference halftone, and find the translation parameters that minimize the difference in the sum square error (SSE) between them over a small subset of image locations. In practice, we use 100 image points arranged on a 10x10 grid. In Figure 9, the very low frequency filtered components are obtained using a box filter that has been computed efficiently using the equivalent of an integral image (requiring a fixed cost of just 4 additions a multiplication and a shift per pixel [11]) where the box filter of the reference image is 31×31 pixels square, and has been applied to the reference image in advance. The central region of the captured image is filtered in real time using a $31S \times 31S$ box filter where S is the scale factor recovered from the DFT either using the mean of the horizontal and vertical estimates from the previous section or directly or from the affine transform as:

$$S = \frac{\sqrt{a_1^2 + a_2^2} + \sqrt{a_3^2 + a_4^2}}{2}.$$

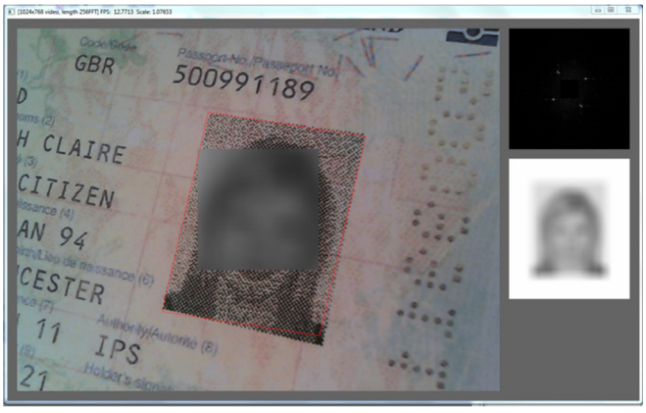

Figure 9. *Location of target as indicated by red outline.*

The gradient decent approach to solving image registration under various classes of transformation uses the Gauss-Newton method and was originally proposed by Lucas and Kanade [12] and has been refined subsequently [13]. Applying such a low pass filter prevents the search process from getting stuck in a local minima and allows registration to succeed even when the starting point is a far from optimal. For improved robustness, parameters to normalize the intensity differences between the images in the optimization process are estimated separately and directly from the images and the current estimate of the translation and affine parameters. The image intensity parameters (scale and offset) are only updated each time the optimization finds a local minimum; the method continues until no further improvements are possible or a maximum iteration count is exceeded.

The filtered version of the central 256x256 region of the captured image is used as the reference image plus a border half the size of the box filter $(31S/2)$. The iterative optimization locates this image region within the target filtered "reference halftone" image from which the position of the stegatone (which has the same dimensions as the reference halftone) within the captured image can be estimated and is shown as the red outline in Figure 9.

Once the location of the Stegatone is stable within the capture frame video sequence, its location in the current frame can be finalized. Starting from the affine estimate the parameters of a 3x3 homogeneous transformation G that represents the planar projection (or homography) are computed using a second application of the Lucas and Kanade method, i.e.,

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} A & t \\ v^T & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix},$$

or simply $\mathrm{x}' = G\mathrm{x}$, where finally $x'' = x'/w'$ and $y'' = y'/w'$. Note that the planar homography is the most general form of this transformation and can be considered a combination of the affine A and translation t components when the elements of v are zero.

In this step, we construct a multi-scale band-pass representation of the halftone reference and captured images similar in spirit to the approach of Bouguet [14], and use a larger number of image locations (20x20) in the optimization. Again, for the reference halftone this information can be computed in advance, and for the captured image limited to a region that includes the estimated location of the stegatone plus a modest border (in our experiments 100 pixels). As the affine plus translation estimate of the transform is already demonstrates good performance,, we have found that the multi-scale representation only needs two levels and is constructed by successive application of Gaussian filters (3 times) where the effective standard deviation (σ) of the filter doubles at each level. We find a value of σ=3.0 for the base level (and hence σ=6.0 and σ=12.0 for the subsequent levels) to be effective (though for the captured image they are modified to take into account the scaling factor recovered by the DFT). The two band-pass levels are achieved by subtracting successive pairs of Gaussian filtered images to give the final multi-scale Difference of Gaussian representation, as shown in Figure 10.

In general, if an image has been pre-filtered by a Gaussian of size σ and we wish to achieve an effective smoothing of size $s\sigma$ (where s is a scale factor) then the additional Gaussian filter that

needs to be applied is of size $\sigma\sqrt{s^2-1}$ or in this case where the scale factor is two $\sigma\sqrt{3}$ which reduces the computational cost by 15%. As the Gaussian is separable it is computed as two one-dimensional filters and thus the speed improvement is linear. In all cases we represent our images and their filtered counterparts using 16 bit integer arithmetic.

An example of the final determined location including the perspective distortion is shown overlaid as a red quadrilateral in Figure 10. Note that we do not align the reference image itself as part of the multi-scale approach, only the band-pass versions. The reference halftone image is of course highly idealized and is geometrically different from the printed and captured stegatone derived from it, since in the latter many of the individual dot locations have been changed to encode information. As a result, there is no advantage in terms of transform accuracy in proceeding down to register the image details themselves.

Figure 10. *Successive difference-of-Gaussian representations of reference halftone (top), and captured stegatone (bottom).*

Mobile Performance

The detection system was implemented on both Android and iOS phones. Using a 256x256 video capture window the clients can perform DFT-based scale and rotation determination at 20 frames per second. Example views of our UI (on an iPhone) are shown in Figure 11. An overlay of a red rectangle indicates that the object is not resolved; in the case on the left, the camera distance is closer than 8 cm and thus unable to focus. The overlay turns green when the characteristic periodic structure is clearly evident and correctly positioned in the DFT, as in the case on the right. Only images that are within the correct scale range are analyzed for full perspective alignment.

This initial part of the processing can be carried out close to the frame-rate of the sensor even on a low powered mobile device, but in this system, the detected image is uploaded to a server for full perspective alignment. The fine scale alignment depends on the size of the Stegatone and its scale within the image but for the example shown in Figure 9 this alignment is typically achieved in less than 0.2 seconds on an 8 core processor. This stage can clearly take place on a mobile client with some increase in processing time, but must be fast enough to provide smooth interaction with a user.

Figure 11. *Mobile UI of fast stegatone detector. Screen captures from an iPhone 5S.*

We also tested the data recovery rates for mobile captured images on 76 small passport "ghost" images of size 15x10 mm, representing faces from a wide range of ethnicities and ages. Our hand-held tests varied the capture distance, rotation angle and obliqueness for each of these images. The results were surprisingly robust with an average recovery rate of 90.5% for the 256 encoded bits. Further optimization of the effective modulation and error-protection schemes for video capture will enable improvements upon this result.

As a final example that illustrates the versatility of this system, consider the circular logo in Figure 12. Even though there are no rectangular edges to find, the clear peaks in the DFT (Figure 13) allow for fast and accurate detection of scale and orientation. Based on the peak locations the system finds a rotation angle of 24.9°, a stegatone cell size of h=7.55 and v=7.51 capture pixels, and a capture resolution of C_h=755.4 by C_v=750.6 dpi. While we demonstrated that this DFT-based detection scheme is well suited for locating stegatones, it works equally well for any 2D periodic or quasi-periodic pattern.

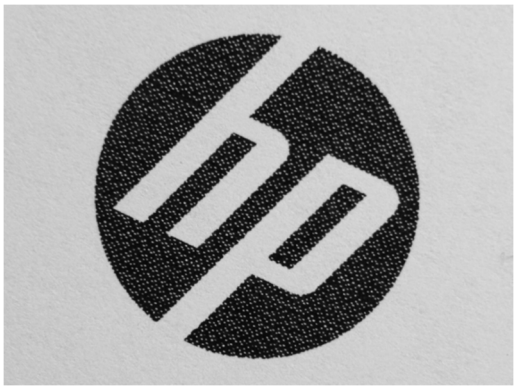

Figure 12. *Mobile capture of a non-rectangular-shaped stegatone.*

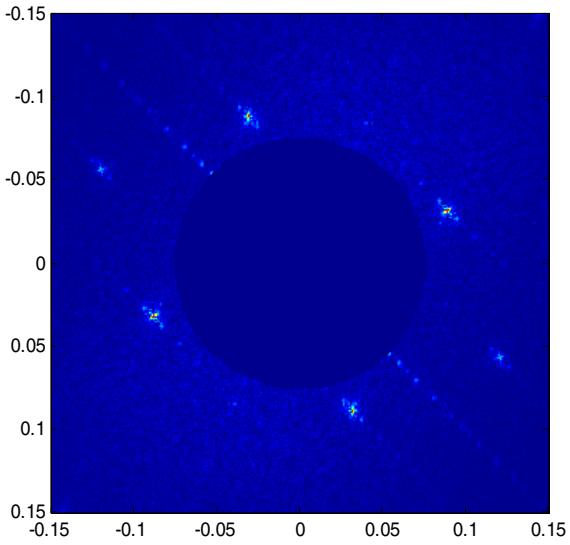

Figure 13. *DFT of the captured image in Figure 12.*

References

[1] R. Ulichney, M. Gaubatz, and S. Simske, “Encoding Information in Clustered-Dot Halftones”, IS&T NIP26 (26th Int. Conf. on Digital Printing Technologies), Austin, TX, 602-605, Sep 2010.

[2] O. Bulan, V. Monga, G. Sharma, and B. Oztan, “Data embedding in hardcopy images via halftone-dot orientation modulation,” in Proc. SPIE: Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, 6819, pp. 68190C-1-12, Jan. 2008.

[3] J. Lundström, A. Verikas, "Detecting halftone dots for offset print quality assessment using soft computing," Fuzzy Systems (FUZZ), 2010 IEEE International Conference on , vol., no., pp.1,7, 18-23 July 2010.

[4] R. Vander Kam, R. Gray, "Lossy compression of clustered-dot halftones using sub-cell prediction," Data Compression Conference, 1995. DCC '95. Proceedings , vol., no., pp.112,121, 28-30 Mar 1995.

[5] X. Duan, G. Zheng, H. Chao, "An adaptive real-time descreening method based on SVM and improved SUSAN filter," Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on , vol., no., pp.1462,1465, 14-19 March 2010.

[6] P. Viola and M. Jones, “Robust Real-time Object Detection,” ICJV, 2001.

[7] D. G. Lowe, “Object Recognition from Local Scale-Invariant Features,” Proceedings of the International Conference on Computer Vision 2, 1999.

[8] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool “SURF: Speeded Up Robust Features”, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, 2008.

[9] R. Ulichney, Digital Scaling of Binary Images, M.S. Thesis, M.I.T., Cambridge, MA, ch 4, 1979.

[10] R.N. Bracewell, K.-Y. Chang, A.K. Jha, Y-H. Wang, “Affine theorem for two-dimensional Fourier transform”, Electronics Letters, 29(3): 304, 1993.

[11] J. Wojciech, “Fast image convolutions,” in ACM SIGGRAPH workshop, Illinois, USA, 2001.

[12] B. Lucas, T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision”, DARPA Image Understanding Workshop, 121-130, 1981.

[13] S. Baker and I. Matthews, “Lucas-Kanade 20 Years On: A Unifying Framework”, Intl. Jnl. of Computer Vision, 56(3), 221-255, 2004.

[14] J-Y. Bouguet, “Pyramid Implementation of Lucas Kanade Feature Tracker: Description of the algorithm”, OpenCV Documents, Intel Corporation, 1999.

Author Biography

Robert Ulichney is a Distinguished Technologist with HP Labs. He received a Ph.D. from MIT in electrical engineering and computer science. Before joining HP he was with Digital Equipment Corp for several years then with Compaq’s Cambridge Research Lab where he led a number of research projects on image and video implementations for both hard copy and display products.