https://doi.org/10.2352/ISSN.2470-1173.2020.4.MWSF-024


# Estimating Digital Watermark Synchronization Signal Using Partial Pixel Least Squares

***Robert Lyons and Brett Bradley, Digimarc Corporation, 9405 SW Gemini Drive, Beaverton, OR, USA 97008-7192***

## Abstract

*To read a digital watermark from printed images requires that the watermarking system read correctly after affine distortions. One way to recover from affine distortions is to add a synchronization signal in the Fourier frequency domain and use this synchronization signal to estimate the applied affine distortion. If the synchronization signal contains a collection of frequency impulses, then a least squares match of frequency impulse locations results in a reasonably accurate linear transform estimation. Nearest neighbor frequency impulse peak location estimation provides a good rough estimate for the linear transform, but a more accurate refinement of the least squares estimate is accomplished with partial pixel peak location estimates. In this paper we will show how to estimate peak locations to any desired accuracy using only the complex frequencies computed by the standard DFT. We will show that these improved peak location estimates result in a more accurate linear transform estimate. We conclude with an assessment of detector robustness that results from this improved linear transformation accuracy.*

## Introduction

A blind watermarking system, as described in Cox [3], must recover watermarks after geometric distortions without access to the original cover image. Affine distortions, which include scaling, rotations, cropping and translation distortions, are the simplest and most important geometric distortions and this paper will focus on them. More complex non-linear geometric distortions, often found in camera capture, include projective transforms or spherical aberrations. If the watermark is printed and then captured with a scanner or smart phone, then estimating and recovering from geometric distortions becomes an acute problem.

When a watermarked image is printed and captured, the resulting image is distorted. This distortion is usually close to an affine transform when restricted to a small region. An affine mapping has the form,

$$A(x) = M(x) + t, \tag{1}$$

where $x \in R^2$, $M$ is a $2 \times 2$ matrix and $t$ is a translation 2-vector. Reading a watermark from this captured image requires estimating this affine transform so that the watermark extraction process is aligned properly.

We know from previous work, such as Alattar [1], that a synchronization signal in the frequency domain can be used to correct many forms of geometric distortions, especially affine distortions. For example, one can insert a set of frequency impulses, which are just corrugations in the spatial domain. These frequency impulses provide a good mechanism for estimating the linear transform as frequency magnitudes are independent of the translation. Therefore, one can use a variant of the least squares method on frequency magnitudes to estimate the linear transform portion of the affine transform. If we apply an affine transform, $A(x) = M_s(x) + t$, in the spatial domain, then the frequency impulses will transform with the linear transform $M_f = \left(M_s^T\right)^{-1}$. This means that the linear transform in the frequency domain gives us the linear transform in the spatial domain.

To see how the synchronization signal fits into a watermarking system, we give an example of a detector that detects a watermark signal with an added synchronization signal that consists of a collection of Fourier Domain impulses.

1. Take the DFT of the entire image or a selected region of the image.
2. Find the synchronization signal Fourier domain impulses. We shall elaborate on this step in this paper.
3. Find a frequency domain linear transform $M_f$ that maps the original carrier signal peak constellation onto the observed peaks. The linear transform in the spatial domain will be $M_s = \left(M_f^T\right)^{-1}$. This step is described in more detail below.
4. Compute the phases of the synchronization signal in the image block. One method of doing is to use the phase estimation technique proposed in [6].
5. Use the phases of the carrier signal to compute the translation using phase correlation [12] or some equivalent technique.
6. Use the linear transform $M_s$ and the translation offset to align the image to its original framed position.
7. Extract the watermark.

The objective of this research is to improve the accuracy of the linear transform estimation for a watermarking system. The least squares estimate minimizes the squared error between the set of observed impulse locations and coordinate locations resultant from applying a linear transform to the original coordinates. In the discrete Fourier domain these locations are easily measured to integer pixel locations by choosing the frequencies with the largest magnitude. However, the least squares estimate is often more accurate if the impulse locations can be measured to partial pixel accuracy. This additional accuracy will generate a more accurate linear transform that will better predict the geometry of the watermark and so result in a more robust watermark extraction. In this paper we will show how to estimate frequency impulse locations to any desired accuracy. This additional peak location estimation accuracy results in a more accurate linear transform estimate.

We begin the paper with a short review of 2-dimensional least squares. There is some subtlety in how one can match the original synchronization impulses with the peaks observed in the frequency domain. One could use RANSAC [8], Log-Polar [11] or some equivalent method. We used the direct least squares proposed in [12]. We shall not discuss this in detail in the current paper.

After a short review of least squares, this paper discusses an algorithm used to estimate the magnitude and phase of a frequency impulse. In watermarking, the image is considered noise and so we are searching for a small signal in a large amount of background noise. Further, we only consider algorithms that have straightforward hardware implementations. There are several algorithms proposed in the literature. Our implementation is the same as the implementation in [6]. There are other algorithms that estimate the frequency magnitude including [9], [10], [5]. Our method assesses the magnitude and phase for a given frequency location.

We conclude the paper by describing how this additional peak search accuracy fits into a watermarking system. We present results for a test harness and for an actual watermarking system.

## Least Squares

To estimate the linear portion of affine distortion we use least squares in the frequency domain. To execute least squares you must find the locations of the original synchronization signal impulses after the geometric distortion. Assume we have original impulse locations given by points $(x_i, y_i) \in R^2$ with $i = 1, \cdots S$. Assume we find associated locations given by the measurements $(f_i, g_i)$ which are just the location of the impulse peaks in the actual image. In the introduction we discussed some of the subtleties of this association. We want to find the best fit linear mapping $M : R^2 \to R^2$, with components $m_{ij}$, that maps $(x_i, y_i)$ to $(f_i, g_i)$. If we minimize the squared error, then we end up with a linear regression problem that is easily solved (for details see [13]). The squared error is,

$$e^2 = \sum_{i=1}^{S} \left| \left[ \begin{array}{cc} m_{11} & m_{12} \\ m_{21} & m_{22} \end{array} \right] \left[ \begin{array}{c} x_i \\ y_i \end{array} \right] - \left[ \begin{array}{c} f_i \\ g_i \end{array} \right] \right|^2 \tag{2}$$

To find the matrix that minimizes this we set the partials to zero,

$$\frac{\partial e}{\partial m_{ij}} = 0.$$

This yields four equations which result, after a bit of algebra, in the following solutions,

$$\begin{array}{l} \left[ \begin{array}{c} \hat{m}_{00} \\ \hat{m}_{01} \end{array} \right] = \frac{1}{\det(L)} L \left[ \begin{array}{c} \sum_i f_i x_i \\ \sum_i f_i y_i \end{array} \right] \\ \left[ \begin{array}{c} \hat{m}_{10} \\ \hat{m}_{11} \end{array} \right] = \frac{1}{\det(L)} L \left[ \begin{array}{c} \sum_i g_i x_i \\ \sum_i g_i y_i \end{array} \right] \end{array} \tag{3}$$

where the $2 \times 2$ linear transform $L$ is given by,

$$L = \left[ \begin{array}{cc} \sum_i y_i y_i & -\sum_i y_i x_i \\ -\sum_i x_i y_i & \sum_i x_i x_i \end{array} \right].$$

The starting impulse locations $(x_i, y_i)$ are known and the $(f_i, g_i)$ are found on the distorted image. The more accurately we know the position of the impulse on the distorted image the more accurately we can estimate the updated linear transform components $\hat{m}_{ij}$.

Searching for impulse locations using nearest neighbor is a reasonable way to find the impulse locations $(f_i, g_i)$. But we can improve this by estimating the impulse locations to partial pixel. Center of mass techniques yield better results than nearest neighbor but do not use the phase of the frequency values. In subsequent sections we will show how to find the impulse locations using the full complex values in the frequency plane.

## Finite Sampling and Spectral Leakage

We start in 1-dimensions and define a continuous signal, $f : R \to R$. The Fourier Transform of $f$ is the function $\mathscr{F}\{f\} : R \to C$ given by

$$\mathscr{F}\{f\}(\mu) = F(\mu) = \int_{-\infty}^{\infty} f(x) \exp(-2\pi j \mu x)\, dx$$

The Nyquist Sampling theorem says that if $F(\mu) = 0$ for $\mu \notin [-B, B]$ then $f$ can be reconstructed from samples, if they are taken uniformly at a frequency greater than $2B$ (see [7]). Let the sampling frequency be $\nu_s \geq 2B$ and define the sampling interval $\Delta$ by

$$\Delta = \frac{1}{\nu_s} \leq \frac{1}{2B}.$$

One way to justify the Nyquist sampling theorem is to write the function $F(\mu)$ as a Fourier series. Let $F_c$ be the Fourier series expansion of $F$,

$$F_c(\mu) = \sum_{k=-\infty}^{\infty} a_k exp\left(-\frac{2\pi j}{1/\Delta} \mu k\right) = \sum_{k=-\infty}^{\infty} a_k exp(-2\pi j \mu k \Delta).$$

The sign of $k$ is non-standard and is chosen for convenience. The coefficients are

$$a_k = \frac{1}{\nu_s} \int_{-\nu_s}^{\nu_s} F(\mu) exp(2\pi j \mu k \Delta)\, d\mu = \Delta f(k\Delta),$$

by the inverse Fourier Transform formula. We know that $F_c(\mu) = F(\mu)$ whenever $|\mu| \leq \frac{\nu_s}{2}$, however, $F_c$ is periodic, with period $\nu_s$. So, on the interval $[-\frac{\nu_s}{2}, \frac{\nu_s}{2}]$ we can express the Fourier Transform as,

$$F_c(\mu) = F(\mu) = \Delta \sum_{k=-\infty}^{\infty} f[k\Delta] exp(-2\pi j \mu k \Delta).$$

To derive the Nyquist sampling reconstruction formula, you multiply by the indicator function on $\left[-\frac{\nu_s}{2}, \frac{\nu_s}{2}\right]$ and take the inverse Fourier Transform to get the actual signal $f$ (see [7] or [2]).

Our concern is the infinite sum, which is impossible to compute in the laboratory. A real signal must be windowed by a window function $h(x)$ which has compact support. Recall that when we say that a function $h$ has compact support then there is a number $W$ such that $h(x) = 0$ whenever $|x| > W$. Now define the indicator function $1_A(x) = 1$ if $x \in A$ and 0 otherwise. If we take the indicator function on the half-open interval $[0, N\Delta)$, which we denote by $h(x) = 1_{[0,N\Delta)}(x)$, then the windowed function $hf$ has compact support and has just $N$ samples,

$$f(0), f(\Delta), \cdots f((N-1)\Delta).$$

We exclude the point $N\Delta$, since this sample would represent the interval $[N\Delta, (N+1)\Delta)$. Our windowed function has the Fourier Transpose,

$$F_h(\mu) = \Delta \sum_{k=0}^{N-1} f[k\Delta] exp(-2\pi j \mu k \Delta).$$

The Nyquist condition is that the Fourier Transform must have compact support. If the window has compact support, then windowed function $hf$ has compact support as well. But this means that the Fourier Transform cannot have compact support and so cannot satisfy the basic Nyquist criteria. Another, related issue is that the frequency peaks are blurred by the Fourier Transform of the window function. One way to see this is to use the product rule of Fourier Transforms,

$$\mathscr{F}\{hf\} = \mathscr{F}\{h\} * \mathscr{F}\{f\}.$$

This is often called window spectral leakage. See [4] for a good discussion of these issues. We will get an explicit expression below that will help us calculate leakage due to windowing.

Divide the interval $[-\frac{\nu_s}{2}, \frac{\nu_s}{2}]$ into $N$ equal frequency intervals each with width given by $\nu_0$,

$$\nu_0 = \frac{\nu_s}{N} = \frac{1}{N\Delta}.$$

We re-write $F_h$ with frequency coordinate $w$, defined by $w\nu_0 = \mu$. Using $w$, the Fourier Transform of the windowed samples has the form,

$$F_h(w) = \Delta \sum_{k=0}^{N-1} f[k\Delta] exp\left(-\frac{2\pi j}{N} wk\right). \tag{4}$$

If $w$ is an integer then this is the standard DFT, but we do not assume that $w$ is an integer. However, $w$ does satisfy $|w| \leq \frac{N}{2}$. Now move to 2-dimensions and derive an explicit formula for window spectral leakage.

## Spectral Leakage in Two Dimensions

We switch to 2-dimensional blocks. Given a 2-dimensional signal $f : R^2 \to C$ which satisfies the Nyquist condition. Let $\Delta_1$ and $\Delta_2$ be the sampling intervals of the $x$ and $y$ directions respectively. We apply a window $h$ that has $N_1$ non-zero samples in the x-direction and $N_2$ samples in the y-direction. This section computes the window leakage in this 2-dimensional case.

We start with a definition of the vector space structure on the set of sampled functions. Let $L^2(Z_{N_1} \times Z_{N_2})$ be the $N_1N_2$ dimensional vector space consisting of finite complex functions on the set $\{0, \cdots, N_1 - 1\} \times \{0, \cdots, N_2 - 1\}$. A sampled function takes every pair of coordinates $(k_1, k_2)$ to a complex number $f[k_1, k_2]$. We define an inner product and a norm,

$$\langle f, g \rangle = \sum_{k_1=1}^{N_1-1} \sum_{k_2=1}^{N_2-1} \overline{f[k_1, k_2]} g[k_1, k_2]. \tag{5}$$

$$\|f\|^2 = \langle f, f \rangle \tag{6}$$

Note the function $(k_1, k_2) \to f(k_1\Delta_1, k_2\Delta_2)$ is in this space.

We want to write the 2-dimensional analog to Equation 4. But we scale magnitudes so that $\Delta_1\Delta_2 \to \frac{1}{\sqrt{N_1 N_1}}$ and adjust the scale domain of $f$ so it is just an indexed object. With this, the equivalent to Equation 4 for the signal $hf$ is given by,

$$F_h(w_1, w_2) = \frac{1}{\sqrt{N_1 N_2}} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} f[k_1, k_2] \exp\left(-2\pi j \left(\frac{w_1 k_1}{N_1} + \frac{w_2 k_2}{N_2}\right)\right) \tag{7}$$

When $u_1$ and $u_2$ are integers then $F_h(u_1, u_2)$ is the DFT of the windowed signal $f_h = hf$,

$$\mathscr{D}\{f_h\}[u_1, u_2] = F_h(u_1, u_2). \tag{8}$$

We choose to scale by $\frac{1}{N_1 N_2}$ for convenience. With this scale the DFT Is an isometry with respect to the norm in 6, meaning, for every $f$ we have,

$$\|\mathscr{D}\{f\}\| = \|f\|.$$

Next, we find the Fourier transform of a pure frequency that is windowed, and so has samples,

$$p_v[k_1, k_2] = \frac{1}{\sqrt{N_1 N_2}} \exp\left(2\pi j \left(\frac{v_1 k_1}{N_1} + \frac{v_2 k_2}{N_2}\right)\right), \tag{9}$$

where $v = (v_1, v_2)$. Note that this function always has norm $\|p_v\| = 1$, with the norm defined in Equation 6. The frequencies, $F_h$, of this signal are

$$\begin{aligned} F_h(w_1, w_2) &= \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \exp\left(2\pi j \left(\frac{(v_1 - w_1)k_1}{N_1} + \frac{(v_2 - w_2)k_2}{N_2}\right)\right) \\ &= P_{N_1}(v_1 - w_1) P_{N_2}(v_2 - w_2). \end{aligned}$$

where,

$$P_N(\delta) = \frac{1}{N} \sum_{k=0}^{N-1} \exp\left(2\pi j \left(\frac{\delta}{N} k\right)\right) = \frac{1 - \exp(2\pi j \delta)}{N\left(1 - \exp\left(2\pi j \left(\frac{\delta}{N}\right)\right)\right)}$$

The sin is the difference of exponentials, so we arrive at,

$$P_N(\delta) = \exp\left(\pi j \delta \left(\frac{N-1}{N}\right)\right) \frac{\sin(\pi\delta)}{N \sin\left(\pi \frac{\delta}{N}\right)} \tag{10}$$

Notice that (in the limit) $P_N(0) = 1$. Notice, also, that as $N \to \infty$ this function looks like a standard *sinc* function. Putting this all together we get,

$$F_h(w_1, w_2) = P_{N_1}(v_1 - w_1) P_{N_2}(v_2 - w_2). \tag{11}$$

which exhibits the Frequency content of a pure frequency impulse that is sampled and windowed with an indicator function. It's worth emphasizing that $P_N$ is a direct result of using the indicator function for a window.

If we construct an image composed from a pure sinusoid, then it is the sum of two signals of the form of Equation 9. Again, we do not assume that $v_1, v_2$ are integers. To simplify the notation we will examine the complex sinusoid,

$$s[k_1, k_2] = Ae^{2\pi j\theta} p_v[k_1, k_2].$$

If we construct the DFT we get, for integer $u_1, u_2$,

$$\begin{aligned} \mathscr{D}\{s\}[u_1, u_2] &= Ae^{2\pi j\theta} \mathscr{D}\{p_v\}[u_1, u_2] \\ &= Ae^{2\pi j\theta} P_{N_1}(v_1 - u_1) P_{N_2}(v_2 - u_2). \end{aligned}$$

In the next section we will use this to estimate $A$ and $\theta$.

## Impulse Detection

In this section we will estimate the magnitude and phase of a frequency impulse using the standard DFT data, without assuming that the frequency has integral coordinates. For a different presentation of this same material, which is based on maximizing signal-to-noise, see [6].

We start with a corrugation, which is a scaled pure frequency. An image might have a cosine wave which is a sum of two complex exponentials and so contains a frequency and its negative. We will use a signal $s$ that is a scaled version of a single frequency impulse, as in Equation 12,

$$s[k_1,k_2] = A\exp(2\pi j\theta)\, p_v[k_1,k_2]. \tag{12}$$

Here $v = (v_1, v_2)$ is not assumed to have integer coordinates and $\theta$ is the phase, normalized to lie in $[-1/2, 1/2]$. We will assume that the signal, at least locally, looks like the impulse of Equation 12. When we find the synchronization peaks, we will assume that they are larger than the neighboring pixel image data.

Take the standard DFT of the signal $s$,

$$\mathscr{D}\{s\}[u_1,u_2] = S[u_1,u_2] =< p_{u_1,u_1}, s > .$$

This means that a pure signal, with no noise, satisfies

$$S[u_1,u_2] = Ae^{2\pi j\theta} < p_{u_1,u_2}, p_v > . \tag{13}$$

We know, from definition, that $\mathscr{D}\{p_v\}[u_1,u_2] =< p_{u_1,u_2}, p_v >$. With this we have,

$$< p_{u_1,u_2}, p_v >= P_{N_1}(v_1-u_1)P_{N_2}(v_2-u_2). \tag{14}$$

We want to estimate $A$ and $\theta$ so we form the expression,

$$S[u_1,u_2]\,(< p_{u_1,u_2}, p_v >)^* = Ae^{2\pi j\theta}\,\| < p_{u_1,u_2}, p_v > \|^2 .$$

Now sum up the values on all possible integer frequency locations and we get the expression,

$$\begin{aligned}\sum_{u_1=0}^{N_1-1}\sum_{u_2=0}^{N_2-1} S[u_1,u_2] \quad & (P_{N_1}(v_1-u_1)P_{N_2}(v_2-u_2))^* \\ &= \sum_{u_1=0}^{N_1-1}\sum_{u_2=0}^{N_2-1} Ae^{2\pi j\theta}\,|< p_{u_1,u_2}, p_v >|^2 \\ &= Ae^{2\pi j\theta}\sum_{u_1=0}^{N_1-1}\sum_{u_2=0}^{N_2-1} |< p_{u_1,u_2}, p_v >|^2 \\ &= Ae^{2\pi j\theta}\|p_v\|^2.\end{aligned}$$

where we used Parseval's identity (which in this case is really just the Pythagorean theorem). Recall that the collection of $N_1N_2$ functions $p_{u_1,u_2}$ with $0 \le u_1 < N_1, 0 \le u_2 < N_2$ actually form an orthonormal basis of $L^2(Z_{N_1} \times Z_{N_2})$ (for more on this see [2]). We chose the scaling of $p_v$ so that $\|p_v\|^2 = 1$. We arrive at,

$$\sum_{u_1=0}^{N_1-1}\sum_{u_2=0}^{N_2-1} S[u_1,u_2]\,(< p_{u_1,u_2}, p_v >)^* = A(v)e^{2\pi j\theta(v)}. \tag{15}$$

This gives an estimate for $A(v)$ and $\theta(v)$ based on our frequency impulse point spread function given in Equation 14. We emphasize that the estimates for $A$ and $\theta$ are for the specific frequency location $v = (v_1, v_2)$. Equation 15 is the same equation derived in [6], but it was derived through different means.

To estimate the magnitude and angle we need to take enough terms. Exactly how many terms you need depends on the signal and the carrier. To help find the number of terms required recall that $p_{u_1,u_2}$ form an orthonormal basis on our vector space $L^2(Z_{N_1} \times Z_{N_2})$. This means that,

$$1 = \sum_{u_1=0}^{N_1-1}\sum_{u_2=0}^{N_2-1} |< p_{u_1,u_2}, p_v >|^2. \tag{16}$$

So, for a fixed $v$ we choose as many coefficients as we need to so that the remaining elements $|< p_{u_1,u_2}, p_v >|^2$ are small. This will limit the size of the terms involving $< p_{u_1,u_2}, p_v >^*$. For almost all applications a handful of terms will suffice.

We can evaluate the system when the original block has added white Gaussian noise $N_s$, with standard deviation $\sigma$. The DFT of AWGN, which we denote by $N_f$, is AWGN on each of the real and imaginary parts of the complex frequency. The noise $N_f$ on the real and imaginary parts is independent and has standard deviation $\frac{\sigma}{\sqrt{2}}$. It is straightforward to show that, $< f, N >$ is Gaussian and satisfies,

$$E\left[|< f, N_f >|^2\right] = \|f\|^2 E\left[N_f^2\right] = \|f\|^2\frac{\sigma^2}{2}. \tag{17}$$

So AWGN in the spatial domain corresponds to AWGN in the expansion 15. Equation 17 gives one a way to characterize the variance after the number of terms required for the application is fixed.

### *Zero Padding DFT*

We can take another approach to computing frequencies at non-integer coordinates. In this section we zero pad the image and then perform a longer DFT. This results in comparable equations.

We take our $N$ samples and pad $N$ zero samples to get a $2N$ sample signal $f$. How does the DFT relate to the original DFT and to our frequency interpolation? The $2N$-DFT, $F_{2N}$, is,

$$\begin{aligned}F_{2N}[v] \quad &= \tfrac{1}{\sqrt{2N}}\sum_{k=0}^{2N-1} f[x]\exp\left(-\tfrac{2\pi j}{2N}vk\right), \\ &= \tfrac{1}{\sqrt{2N}}\sum_{x=0}^{N-1} f[x]\exp\left(-\tfrac{2\pi j}{N}\tfrac{v}{2}k\right),\end{aligned}$$

where $v = 0, \cdots 2N-1$. Notice that we have,

$$F_{2N}[v] = \frac{1}{\sqrt{2}}F_N\left[\frac{v}{2}\right]. \tag{18}$$

This is the 1-dimensional analog to Equation 7 where we allow the frequencies to take on integral and half-integral values. The DFT is invertible so we have,

$$f[k] = \frac{1}{\sqrt{N}}\sum_{u=0}^{N-1} F_N[u]\exp\left(\frac{2\pi j}{N}uk\right).$$

Use this to write $F_{2N}[u]$ in terms of $F_N[u]$,

$$\begin{aligned}F_{2N}[v] \quad &= \tfrac{1}{N\sqrt{2}}\sum_{u=0}^{N-1} F_N[u]\sum_{x=0}^{N-1}\exp\left(\tfrac{2\pi j}{N}\left(u-\tfrac{v}{2}\right)k\right) \\ &= \tfrac{1}{\sqrt{2}}\sum_{u=0}^{N-1} F_N[u]P_N(u-\tfrac{v}{2}) \\ &= \tfrac{1}{\sqrt{2}}\sum_{u=0}^{N-1} F_N[u]\left(P_N(\tfrac{v}{2}-u)\right)^*\end{aligned}$$

Insert Equation 18 into this expression and we arrive at,

$$F_N\left[\frac{v}{2}\right] = \sum_{u=0}^{N-1} F_N[u]\left(P_N\left(\frac{v}{2}-u\right)\right)^* \quad (19)$$

This is a 1-dimensional version of Equation 15 when we use 14. Zero padding provides similar information but is more difficult to compute as it requires one to recompute a longer DFT which is quite expensive. To compute values $F_N\left[\frac{v}{2}\right]$ for half-integral frequencies requires a DFT of length $2N$. Other values of $v$ require similarly expensive computations with longer DFT's.

## Partial Pixel Least Squares

To improve a linear transform estimate using least squares we must estimate the frequency peak locations $(f_i, g_i)$. The linear transform estimate will minimize the error described in Equation 2. To start this coordinate update process, we apply the starting linear transform to the synchronization peak coordinates. To determine the updated synchronization signal coordinates, we search for a peak in the neighborhood around the transformed synchronization signal coordinates. Fix a collection of frequency coordinates centered at the transformed coordinates,

$$(v_{1,1}, v_{1,2}), (v_{2,1}, v_{2,2}), \cdots, (v_{m,1}, v_{m,2}).$$

Now use Equation 15 to estimate $A(v_{k,1}, v_{k,2})$ and $\theta(v_{k,1}, v_{k,2})$. We do this for each of the $m$ coordinates and choose the coordinate $k$ with the largest $A(v_{k,1}, v_{k,2})$. After finding the maximum $A(v_{k,1}, v_{k,2})$ we set,

$$\begin{aligned} f_i &= v_{k,1} \\ g_i &= v_{k,2}. \end{aligned}$$

We repeat this for every synchronization point $i = 1, \cdots, S$. After estimating all the peak locations, $(f_i, g_i)$, we can evaluate a new linear transformation using Equation 3.

## Results

We constructed a test harness to evaluate the least squares linear transform accuracy. The data for the model consists of a synchronization signal with added noise,

$$im[k_1, k_2] = \sum_{i=1}^{K} A_i \cos\left(2\pi\left(\frac{v_1(i)k_i}{N_1} + \frac{v_2(i)k_2}{N_2}\right)\right) + N[k_1, k_2],$$

where $N[k_1, k_2]$ is added white Gaussian noise. We tested the linear transform accuracy for whole and $\frac{1}{8}$ pixel accuracy. This means that, for the $i$th frequency impulse, we evaluated $A(v_1(i), v_2(i))$ for every point on a $\frac{1}{8}$ pixel grid near the integer location with the maximum frequency. The result is a linear transformation that is more accurate than the nearest neighbor equivalent. In the results that follow we used,

- Signal amplitude $= 0.29$.
- Additive White Gaussian Noise with $\sigma = 4.0$.

In figure 1 we show the peak search for a single synchronization frequency impulse. The peak location is not exact because of the added noise.

In figure 2 we show the resulting increase in accuracy for a collection of rotations. The improved accuracy is demonstrated by the lower standard deviation of error,

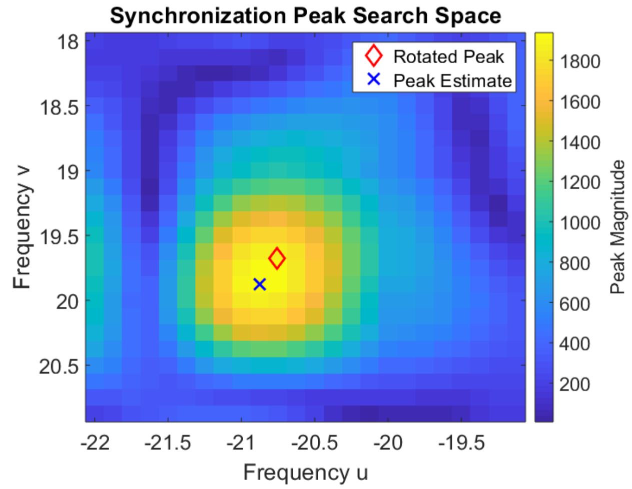



***Figure 1.*** *Search Space for Synchronization Peak*

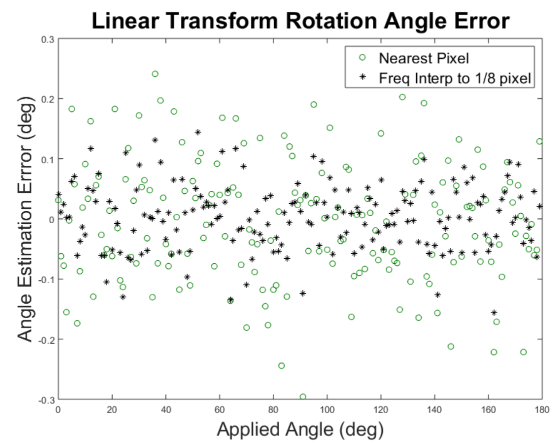



***Figure 2.*** *Linear Transformation Accuracy*

**Test Harness: Standard Deviation of Error**

| Test Harness | Error: Standard Deviation |
|---|---|
| Whole Pixel | $\sigma = 0.103$ |
| 1/8 Pixel | $\sigma = 0.054$ |

The actual impact on detector performance depends on the nature of the embedded signal. The linear transformation is used to synchronize the signal. Different watermarking modulation techniques may have different linear transform accuracy requirements. The accuracy is increased by searching on a finer grid. If less accuracy is needed, one can trade off accuracy for speed by searching on a coarser partial pixel grid.

We incorporated partial pixel peak search into a full watermark detector to test the importance of the algorithm in a full system. This particular detector is quite robust, and the partial pixel peak search does help increase this robustness. Note that there is a robustness gain in the perspective test as well. A perspective transform can often be approximated by an affine transform, but the linear transform may have a large amount of differential scale or shear and so is somewhat more difficult to detect. It is not

**Partial Pixel Peak Search: Detector Results**

| Rotation/Scale Tests | Detection Rate |
|---|---|
| Whole Pixel | 95% |
| Quarter Pixel | 96% |
| Perspective Test | Detection Rate |
| Whole Pixel | 84% |
| Quarter Pixel | 86% |

yet clear how the extra linear transform accuracy helps in the perspective case. More details pertaining to perspective mitigation are covered in the Digimarc patent [12].

## Acknowledgments

We'd like to thank all the people at Digimarc who helped prepare and proofread this document including Adnan Alattar, Ravi Sharma and Joel Meyer.

.

## References

[1] A. Alattar, Bridging Printed Media and the Internet via Digimarc's Watermarking Technology, Multimedia and Security Workshop, ACM MM (1998).

[2] Brémaud Pierre, Mathematical Principles of Signal Processing : Fourier and Wavelet Analysis, Springer, New York, NY, 2002.

[3] Ingemar Cox, Matt Miller, Jeffrey Bloom, Jessica Fridrich and Ton Kalker, Digital Watermarking and Steganography, Elsevier, Morgan Kaufmann Publishers, Burlington, MA 01803, 2008.

[4] Fredric J. Harris, On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, Proc. IEEE, vol 66, pp 51-83, Jan 1978.

[5] Eric Jacobsen and Peter Kootsookos, Fast, Accurate Frequency Estimators [DSP Tips and Tricks], IEEE Signal Processing Magazine, vol 24, pp 123 - 125, Jan 2007.

[6] Robert Lyons and John Lord, Estimating Synchronization Signal Phase, Proc. SPIE 9409, Media Watermarking, Security, and Forensics 2015, 94090P (4 March 2015).

[7] Alan Oppenheim, Ronald Schafer, John Buck, Discrete-Time Signal Processing, (2nd Edition), Prentice-hall, Upper Saddle River, NJ 07458, 1999.

[8] S. P. Pun, Fast Robust Template Matching for Affine Resistant Image Watermarks, IEEE Transactions on Image Processing, 1123 - 1129 (2000).

[9] Barry Quinn, Estimating frequency by interpolation using Fourier coefficients, IEEE Transactions on Signal Processing, vol 42, 1264 - 1268 (1994).

[10] Barry Quinn, Estimation of frequency, amplitude, and phase from the DFT of a time series, IEEE Transactions on Signal Processing, vol 45, 814 - 817 (1997).

[11] G. B. Rhoads, Methods for surveying dissemination of proprietary empirical data, US Patent 5,862,260, 1999.

[12] Sharma et al, Signal Processor and Methods for Estimating Geometric Transformations of Images for Digital Data Extraction, U.S. Patent US9,959,587 B2, July 6, 2017.

[13] Larry Wasserman, All of Statistics: A concise course in Statistical Inference, Springer, New York, NY, 2004.

## Author Biography

*Robert Lyons received his BS in mathematics from Reed College (1979), his MA in physics from University of California, Berkeley (1982) and his MS in ECE from Portland State University (2005). He has worked at Digimarc Corporation since 1999.*

*Brett Bradley received his BS in physics from University of California, San Diego (1993), and his MS in ECE from University of Colorado, Boulder (1997). He has worked at Digimarc Corporation since 1999.*

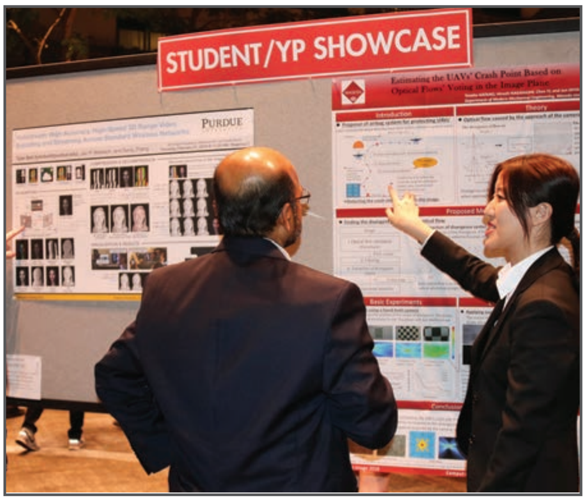
## JOIN US AT THE NEXT EI!

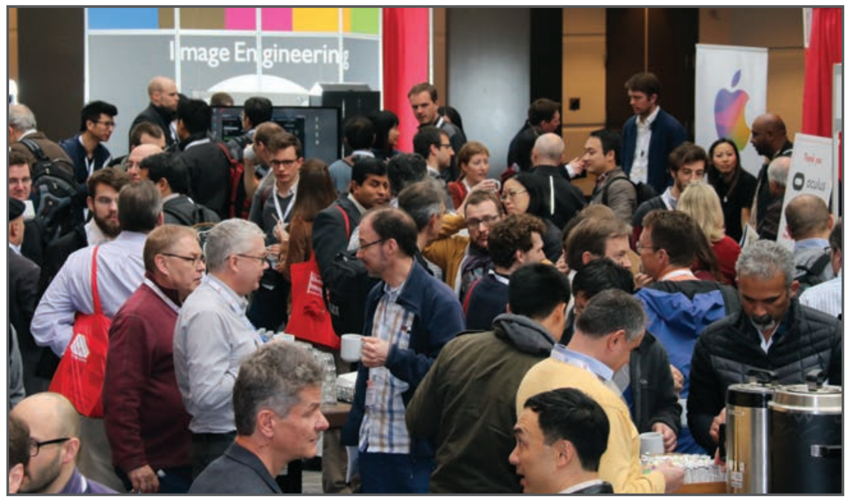
IS&T International Symposium on

# Electronic Imaging

SCIENCE AND TECHNOLOGY

*Imaging across applications . . . Where industry and academia meet!*




**• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •**
**• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •**

**www.electronicimaging.org**

IS&T
imaging.org