A Methodology for Evaluating Accuracy of Capacitive Touch Sensing Grid Patterns
Humza Akhtar and Ramakrishna Kakarala

Abstract—In recent years, there has been significant increase in research conducted in the field of capacitive touch technology. Capacitive touch screen technology, both in consumer electronics (for example smart phones, MP3 players etc) and in wall mount displays (for example ATM kiosks) seem to be omnipresent in the future. This paper discusses the physical design of the self capacitive grid patterns used in current smart phones and the interpolation algorithms used to determine the position of the touch. The main aim of this paper is to simulate and test different touch screen patterns other than the widely used diamond shaped grid design and to provide a methodology for the accuracy comparison of sensor patterns. The paper deals with self capacitance and presents results based on simulations for various sensor patterns.

Index Terms—Capacitive Touch Sensor, Subpixel Interpolation, Grid Patterns, Fourier Analysis

I. INTRODUCTION

Since 1965 [1] inventors have been trying to come up with devices incorporating touch technology. Nowadays almost all smart phone makers have either started producing touch screen smart phones or started doing research in this direction after the trend set by Apple with its release of iPhone [2] in 2007. The world is looking at an explosion of multi-touch display devices (or surfaces) in years to come. These touch sensing technologies differ in the approach they follow in sensing multiple touches and recognizing and interpreting them. Touch technologies can be divided into resistive and capacitive methods. Although expensive to manufacture, capacitive touch screens are more durable and offer multi touch capability with much less complexity in the circuit design as compared with resistive touch technology. This paper only deals with capacitive touch technology.

With the rise of capacitive touch screens into every aspect of our lives, there is an increasing demand of large sized touch screens for various applications in the market. This is a big challenge for the designers and the researchers because as the area increases, so does the noise induced by LCD below the touch panel and due to other environmental factors. For example, a 10.1 inch touch screen panel which may have 51 columns and 29 rows would have 1479 intersections and would require between 200 to 2000 Hz scanning time [4]. This paper probes into grid pattern design of capacitive touch panels and provides a methodology to compare the performance (accuracy) of touch panels so that an optimal design pattern can be constructed.

Capacitive sensors are mostly constructed from Indium Tin Oxide (ITO) which forms the lower conductive coating. When human finger touches the protective layer, it serves as the second conducting layer. This phenomenon is illustrated in Figure 1.

Fig. 1: Phenomenon of capacitive sensing. Blue line represents the lower ITO conductive coating and brown arrow shows a human finger. These two form a capacitive circuit as shown in the dotted line.

The alternating current flowing is continuous across the ITO surface, and when a human body touches the screen, it starts conducting through the body and hence the voltage drop at the touch point is captured as a touch event.

When a touch event is detected, electronic circuits located under the ITO layer measure the distortion produced in voltage and transmit information to the main controller to translate the event into a meaningful gesture [7].

The cost of manufacturing capacitive touch sensing devices (although decreasing with the increase in production volume) is still greater than optical based devices as well as touch resistive technology devices. Because the device must be able to sense changes in capacitance as small as few femtofarads (10^-15 F), therefore the aim of a designer should be finding out the optimal point between the sensitivity of the sensor and its immunity to noise [5]. However capacitive touch sensors have the advantage that they can be integrated in a small area and are thus suitable for mobile phones and PDA devices. Also they are much more responsive than resistive based devices. As this technology responds only to materials which are conductive, a non conductive plastic stylus cannot be used on such screens.

This paper deals with one particular type of capacitive sensing called the self (or absolute) capacitive sensing which is used when the processor wants to determine only one finger touch location over the grid. Although self capacitance can measure multi touch but it is not common practice. As a low cost solution it is widely used in ATM kiosks and in large touch screens used for demonstration and education purposes. Smart phones like iPhone use a different type of touch sensing called mutual capacitive in which the processor can detect
multi touch.

Self capacitance can be understood as a technique in which the processor measures an entire row and column of electrodes for capacitive change [6]. Although this approach can be implemented for one touch or simple two touch gestures, it has limitations for more than two-touch (or complex) gestures. The system may detect two $x$ coordinates $(x_1, x_2)$ and two $y$ coordinates $(y_1, y_2)$ but it has no way of knowing that which $x$ coordinate goes with which $y$ and vice versa. However self-capacitance technology is still very popular in applications which require single touch gestures because of its ease of implementation.

In Figure 2 a self-capacitive system will not be able to tell the difference between two two finger touches shown in red and blue color, but a mutual capacitive system would be able to do so as it measures the potential at each individual node of the panel. In this paper only self capacitance is addressed. However we point out below in discussion section that how the methodology can be adapted to mutual capacitance. As mentioned before self capacitance is useful for low cost solutions like in wall mount displays. Mutual capacitance although expensive is useful for smart phone displays where multi touch is required.

![Fig. 2: Self capacitance vs. mutual capacitance. A self capacitance system will not be able to differentiate between two finger touch $[(x_1, y_2),(x_2, y_1)]$ and $[(x_1, y_1),(x_2, y_2)]$.](image)

Although a tremendous amount of research has been done in the field of capacitive touch technology, there is very little academic literature available on design of grid patterns for capacitive touch screens. Most of the material exists in the form of patents. Companies such as Avago, Synaptics etc. have been designing grid patterns for touch screens and trackpads for more than 10 years [8][9]. Krein and Meadows [10] addressed the issue of optimizing a capacitive panel so as to make it inexpensive and accurate while retaining immunity from electromagnetic interference from the TFT-LCD. They used a quasi-static electric field and applied it to the coating on the panel surface, thus making the computed position independent of coating resistivity. Ruan, Chao and Chen [11] proposed a multi-touch capacitive interface for large size touch panels (more than 12in). They used a low disturbance array cell constructed using four MOSFETs in their sensing circuit in order to eliminate the noise effect from the LCD underneath the touch panel. Tai, Chiu and Chou [12] overcame the same issue of noise reduction by using a single conventional transistor in the touch panel circuit. In order to improve handwriting recognition in capacitive touch screen smart phones, Lin et al. [13] used a passive 3D stylus in order to measure $x$, $y$-directional position and $z$-directional strength during handwriting recognition. Baharav and Kakarala [16] explained the working of touch sensing mechanism of a diamond shaped grid patterns by simulation of a Gaussian finger on a touch panel. The findings presented by them forms the basis of this paper.

The main focus of the paper is to study the grid patterns used by touch screens and to see how the processor determines touch location using subpixel interpolation algorithms. It also presents a methodology to evaluate the performance of grid patterns and studies Fourier analysis of the patterns in order to gain insight into what makes one pattern perform better than the other. A preliminary version of this work was presented in [3].

II. GRID PATTERN DESIGN

The most common pattern used nowadays in almost all of the devices incorporating touch screens is the diamond pattern as shown in Figure 3(d). In this pattern the capacitors are neatly arranged in diamond like structure and are spread over two layers, the upper layer forms the column layer and below it is the row layer. The row and columns are not fused together and thus provide a separate reading of the finger location which is then used by the processor to determine the location of the touch. The basic element of a diamond grid is shown in Figure 3(a).

When a basic element is repeated in both horizontal and vertical direction in two separate layers and then put on top of each other, a whole diamond shaped pattern is formed. It should be noted here that in order to determine the correct position of the finger the processor needs to get readings from both row and column layer. Therefore it first scans the rows and then the columns to get both $x$ and $y$ data which is then passed through an estimation algorithm to determine the location of the finger on the screen. As a rule of thumb the finger area must occupy at least one grid element (row and column both) at a given time in order to get meaningful reading. If the grid basic element (see Figure 3(a)) is greater then the finger diameter itself, it is impossible for the processor to calculate accurate result. Also if the grid element size is reduced in order to get better accuracy, the designer will still end up with a lot of pin outs for even a small size touch screen. Therefore the research problem is formulated as follows.

- To formulate a methodology for calculating the performance of a self capacitive touch screen and try to determine the factors which govern the performance of a capacitive touch system.

It will be an interesting approach to go about changing shape of a capacitive grid pattern basic element so that the new pattern is capable of sensing the finger more accurately even if the size of a basic element is the same or even smaller than the area occupied by the finger. An experiment to investigate this phenomenon is presented in Section IV.

There are quite a few grid patterns proposed by designers over the years besides the diamond pattern. One of these patterns is the interleaved pattern proposed by Harley [8]. The
The main objective of the designer was to devise a grid pattern which will reduce the number of basic elements or electrodes in a capacitive system by interleaving the electrodes which will in turn create a larger region where the finger when sensed by the two adjacent electrodes will result in better interpolation result and also less number of basic elements in the panel. The basic element of the interleaved pattern is shown in Figure 3(b). After repeating the basic element along rows and columns, the resultant full scale touch screen panel is shown below in Figure 3(e).

Tareev [14] filed a patent for a touch pad with interleaved traces. The idea behind the invention was to design a unique layout containing a matrix of traces which are interleaved with each other. The claim was that this pattern will perform better if used as a touch screen because the finger when moving from one trace to the next will be in contact with the interleaved portion of the adjacent trace before leaving the previous trace and thus the reading will be more accurate at all times. This patent also formed the basis of [8].

Mackey [9] shows grid design of a capacitive sensing apparatus in which the shape of the basic element looks like two angled crosses as given in Figure 3(c). However it should be noted that this is not the final form of the angled cross pattern. As the inventor suggests this pattern should not be limited to this shape only but the pattern can be suitably varied while fitting in the basic shape of an angled cross. A 16 x 12 touch panel is obtained by repeating the angled cross along rows and columns as shown in Figure 3(f).

### III. Subpixel Interpolation Algorithms

The research problem presented in this paper is treated as an image processing problem. If finger is treated as a point, it can be observed that a point may occupy more than one pixel as shown in Figure 4. In the case of this paper, the term pixel refers to the grid element.

Fig. 4: A zoomed in image of a point. Notice that a single point covers many pixels.
An approximate model of a point can be a 2-D Gaussian point as its edges are smooth and blurred. Basic equation of such a profile will be

\[
I = I_{\text{background}} + I_{\text{point}} \exp\left(-\frac{(x-u)^2 + (y-v)^2}{2\sigma^2}\right)
\]  \hspace{1cm} (1)

where

- \( I \): intensity
- \((x, y)\): any location in the image.
- \( I_{\text{background}} \): intensity of background.
- \( I_{\text{point}} \): peak intensity of point.
- \((u, v)\): center of point.
- \( \sigma \): amount of spread of the Gaussian.

Here \((u, v)\) give the location of the point. Normal interpolation algorithms can estimate values up to single pixel accuracy however as it can be seen that in order to estimate the position of finger, pixel size accuracy will not give accurate results as the values are in continuous range with subpixel precision. Thus subpixel interpolation algorithms are required.

Fisher and Naidu [15] compared five such subpixel interpolation algorithms namely Gaussian, Center of Mass (COM), Linear, Blais-Rioux and Parabolic while determining the position of a light stripe with subpixel accuracy. They concluded that for estimators which use only three values to calculate the position of the light stripe, the Gaussian algorithm performed the best in terms of accuracy and linear algorithm was the fastest. As Blais-Rioux 4 and Blais-Rioux 8 use four and eight values respectively to determine the position which will in turn consume more computation power, these two algorithms were left out in the simulations for this paper and four out of five algorithms are used. Table I gives the equations used by these estimators whilst estimating the row position of the finger on the screen. It is very important to note here that in the process of estimating the finger location the interpolation algorithms will estimate both row and column readings separately and it is the job of the processor to combine the two estimates. It is assumed here for the sake of simplicity that the finger will occupy more than one basic element of row \((r_i)\) and column \((c_j)\). The maximum value of the row reading is denoted by \(r_i\) whereas \(r_{i-1}\) and \(r_{i+1}\) indicate the reading in the row below and above the maximum respectfully. If we take the case of Gaussian estimator, it will use these three values to calculate estimated location as seen in Table I.

The Gaussian estimator assumes that the observed peak shape of the readings will follow a Gaussian profile. The linear estimator on the other hand assumes that a linear relationship between the spread of values before and after the peak reading [15].

It can be verified [16] that the subpixel estimate value generated by these estimators lies between \(-0.5\) to \(0.5\).

### IV. Simulations and Results

In order to address the problem of providing a methodology for calculating and comparing accuracy of grid patterns, we designed all three grid patterns in MATLAB and passed these panels through a series of tests in order to determine which grid pattern performs better than the others.

The simulated grid patterns used are shown in Figure 3. The panels used in these simulations have 16 rows and 12 columns each. The finger used is shown in Figure 5 and has a diameter of 15mm. It is so because typical dimensions for the row and column spacing (and size) is about 5mm in the touch panels. The basic element is made up of 100 pixels for the sake of simplicity although the number can be changed. Taking the basic element of size 100 pixels will result in a 50um sized single pixel.

![Simulated finger profile used in the simulations. Colour depicting magnitude.](image)

![Finger superimposed on diamond grid. The columns and rows covered by the round finger are highlighted.](image)

### Table I: Subpixel interpolation algorithms

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Equation</th>
</tr>
</thead>
</table>
| Gaussian          | \[
\frac{\ln r_{i-1} - \ln r_{i+1}}{2 \ln r_{i-1} - 2 \ln r_i + \ln r_{i+1}}
\]  |
| Center of Mass    | \[
\frac{r_{i+1} - r_{i-1}}{r_{i-1} + r_i + r_{i+1}}
\]  |
| Linear            | \[
\frac{r_{i+1} - r_{i-1}}{2 r_i - \min[r_{i-1}, r_{i+1}]}
\]  |
| Parabolic         | \[
\frac{r_{i-1} - 2 r_i + r_{i+1}}{2 r_{i-1} - 2 r_i + r_{i+1}}
\]  |
then calculating the increase in capacitance for all the rows and columns one by one. The rows which are covered by the finger will give much higher values as compared to the other rows. These values will be sent to the subpixel interpolation algorithms to estimate the location of the finger.

The response of estimators on all three patterns was tested when finger was moved in diagonal, circular and in vertical direction, thus covering all the possible angles of motion. These tests are explained one by one below.

The diagonal motion experiment was also performed with added Gaussian noise. The main sources of noise are from the LCD display and the charger. As LCD display is tightly coupled with the touchscreen sensor, it can often produce high noise levels (sometimes up to 10V). These noise levels are greater than the finger signal [17]. Similarly, charger noise can exceed touch signals by a large factor. These two sources of noise when combined produce wide range, random and time-varying noise, making accurate noise modeling difficult. As discussed in the introduction, existing research focuses on noise suppression at the analogue front end of the touch estimation process. Our research deals with the digital back end of the touch controller system. Figure 6 shows the effect noise has on back end. In Figure 6 (a) the readings collected by the touch processor in three rows are labelled as x0, x1 and x2. The true peak will lie somewhere between x1 and x2 and is found by applying an algorithm as in Table I. However after addition of noise, the readings change as shown in Figure 6 (b) thus making x2 larger than x1 and in effect changing the peak location. The interpolator has no way of knowing the original maximum value x1 and the interpolation will generate an error.

Figure 7 shows the response of simulating the finger travelling along a straight line at 45°. In Figure 7 and also for the subsequent Figure 8 - 10, the color code is as follows: Blue is the exact route, Red is representing COM, Green is for linear, Black represents Gaussian and Cyan is showing Parabolic estimator result. Figure 7 (g), (h) and (i) shows the same simulation but with the finger diameter reduced from 15mm to 10mm. The basic element of a grid patterns is 5mm in size therefore a finger of diameter 15 mm and 10 mm covers 3 and 2 elements of the grid respectively. Due to decrease in the number of grid elements covered by the finger, the overall mean error is increased. The error of the estimators is given in Table II. It can be easily observed that the interleaved pattern performs better than the diamond and angled cross pattern for all the interpolation algorithms and even when the finger diameter is reduced, the interleaved grid comes on top. This is indicated by bold text cells in Table II.

1) Diagonal movement with additional noise: In this paper, for simulation purposes, we have used Gaussian noise with an amplitude of 0.3 to measure the performance of grid patterns. The results for diagonal movement with added noise are shown in Figure 7 (j), (k) and (l) and in Table II. As seen before, the interleaved pattern performs better than the other patterns.

### TABLE II: Diagonal Movement error comparison. Best values are in bold

<table>
<thead>
<tr>
<th>Interpolator</th>
<th>Diamond Grid</th>
<th>Interleaved Grid</th>
<th>Angled Cross Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean (mm)</td>
<td>Max (mm)</td>
<td>Mean (mm)</td>
</tr>
<tr>
<td>COM</td>
<td>1.0265</td>
<td>1.6503</td>
<td>0.9075</td>
</tr>
<tr>
<td>Linear</td>
<td>0.7723</td>
<td>0.5602</td>
<td>0.6915</td>
</tr>
<tr>
<td>Parabolic</td>
<td>0.1988</td>
<td>0.3889</td>
<td>0.1917</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interpolator</th>
<th>Diamond Grid</th>
<th>Interleaved Grid</th>
<th>Angled Cross Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean (mm)</td>
<td>Max (mm)</td>
<td>Mean (mm)</td>
</tr>
<tr>
<td>COM</td>
<td>0.4012</td>
<td>0.8030</td>
<td>0.3977</td>
</tr>
<tr>
<td>Linear</td>
<td>0.6638</td>
<td>1.3607</td>
<td>0.6065</td>
</tr>
<tr>
<td>Parabolic</td>
<td>0.5968</td>
<td>1.3762</td>
<td>0.5988</td>
</tr>
</tbody>
</table>

### Fig. 6: (a) Touch controller readings with zero noise (b) Touch controller readings with added noise.

#### A. Diagonal movement

The finger was placed at a known location on all three panels and dragged diagonally towards the center of the panel. On every iteration the finger was dragged to another known location on the panel and for each location the algorithm computed the profiles and the stimulated location. The simulated location is then compared with the true location to calculate the error in estimation for each of the four interpolation algorithms. The error is defined as the difference between the true path and the estimated path. Mean error in millimeters was calculated by subtracting the true location and the estimated location in each iteration and then taking the average of the resulting vector at the end of the diagonal motion.

### B. Circular movement

As the diagonal movement only accounts for a 45° angle therefore it was necessary to check if the touch panels perform good with other angles as well. The finger was moved in a circular motion and the corresponding results are shown in Figure 8. As was in the case of diagonal movement, the interleaved pattern performs better than the others.

An angle sweep per every 5° was performed to check the performance of the estimators in terms of deviation from the exact (or true) route in terms of angle and length (see Figure 8 (d), (e) and (f)). The results are presented in Table III. The diamond grid performance is better than the interleaved grid for linear and Gaussian estimators as shown in the table by bold font cell. The error difference is positive in such cases. The angled cross grid has the worst performance of the three.
Fig. 7: Finger moved in a diagonal path. The first, second and third column show the results from interleaved, diamond and angled cross pattern respectively. First row depicts a 7.5mm radius finger with no additional noise. Second row is just a zoomed in version of the first row. In the third row, finger radius is reduced to 5mm. Details of how finger size reduction affects the performance is explained in text. In the fourth row, finger radius is 7.5 mm with the addition of Gaussian noise. Notice that the error in interleaved grid is less than the error generated by the diamond grid and angled cross grid. The units for rows and columns are in pixels. Color code is explained in text. (100 pixels = 5 mm).
Fig. 8: In this figure (a), (b) and (c) show finger moved in a circular path with finger radius: 7.5 mm and (d), (e) and (f) show finger doing an angle sweep with radius 7.5 mm. The first, second and the third column indicate the results obtained from interleaved, diamond and angled cross patterns. The interleaved pattern performs better than the other patterns. Notice that the units for rows and columns are in pixels. Color code is explained in text. (100 pixels = 5 mm).

TABLE III: Circular and angle sweep movement error comparison, the best values are in bold

<table>
<thead>
<tr>
<th>Interpolator</th>
<th>Circular movement with finger radius 7.5mm and zero noise</th>
<th>Angle sweep with finger radius 7.5mm and zero noise</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Diamond Grid</td>
<td>Interleaved Grid</td>
</tr>
<tr>
<td>COM</td>
<td>Mean (mm)</td>
<td>Max (mm)</td>
</tr>
<tr>
<td></td>
<td>0.1118</td>
<td>0.1956</td>
</tr>
<tr>
<td>Linear</td>
<td>0.2429</td>
<td>0.4309</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.0466</td>
<td>0.1172</td>
</tr>
<tr>
<td>Parabolic</td>
<td>0.1931</td>
<td>0.3806</td>
</tr>
</tbody>
</table>

C. Edge effect

As the finger goes near the edge of the screen the performance of the interpolation algorithms gets worse even with no additional noise. This is because of the lack of row and column readings available at the edges of the screen for the interpolation algorithms to perform well. As all of the interpolation algorithms require minimum three readings (see Table I) to calculate an estimate; at the edge the readings reduce to two and then finally to one as the finger continues to move outside the panel. The missing readings were treated as zeroes in the simulation, thus making the results non accurate. In order to reduce the edge effect, a check was imposed and whenever finger reached the edge and there were only two values (A and B) available instead of three, we switched to a more simpler two point linear algorithm.

\[
X = 1 - (0.5) \times \left(\frac{\min(A, B)}{\max(A, B)}\right)
\]

Where \(X\) varies from 0 to 1. The value calculated by this algorithm were used to get a better estimate of the finger movement at the edge. The results are shown in Figure 9.

D. Horizontal, vertical movement and bias of estimators

If the finger is moved straight down or straight up in a vertical fashion it is evident upon careful observation that there is a bias in the readings (see Figure 10). The error induced in the position estimate is due to the fitting algorithms used to track the maximum position. If the position of the finger with respect to the columns are not changed by the user throughout the motion, the fitting algorithms will have same column reading throughout the simulation to track the finger position. This can result in a non uniform fitting curve with a shifted global maxima thus adding a constant bias to the estimate. This type of error is totally different from error due to random noise as it only induces bias in an estimate.
Fig. 9: Minimizing the edge effect on interleaved grid (a) original result. Notice that due to the finger diameter being 300 pixels, the error due to edge effect starts to show after 900 pixels on x-axis when the finger starts leaving the screen (b) after correction. The error starts appearing now from 1000 pixels onwards. color code explained in text.

To see the effect of bias in simulations, the finger starting column position was changed from 450 to 750 in steps of 10 to generate a graph (Figure 11) showing the trend of the bias for all the estimators. The value of bias is obtained by subtracting the mean of the estimator result from the true value.

![Graph showing bias in estimator result](image)

Fig. 10: Bias in estimator result due to the same column value input to the estimators throughout the motion of the finger in the interleaved pattern. Notice that the horizontal axis is scaled to show the bias. Units for rows and columns are in pixels. (100 pixels = 5 mm)

V. DISCUSSION

The diamond pattern is widely used due to the fact that it is very easy to manufacture due to its simplistic design. Touch makers should consider adapting to new patterns so that they can achieve better accuracy. Different companies like Synaptics and Avago Technologies use their own patented designs as explained in Section II of the paper. However the main objective to this research is to compare performance of various patterns with their manufacturing ease.

The methodology discussed in previous sections can be adapted for mutual capacitance systems easily as all that needs to be changed is to find the local maxima in every row and column as well instead of finding out just global maxima. Thus, if a finger covers 3 by 3 grid cell area: In mutual capacitance technique, the processor will create finger reading profile using only 3 cells in the middle row and 3 cells in the

![Graph showing bias in subpixel estimators](image)

Fig. 11: Bias in subpixel estimators. The column value was changed from 450 to 750 in steps of 10 after a single vertical motion by the finger and then mean of the estimator result was subtracted from the true value to get the bias. Notice that linear estimator has the greatest bias thus complying with the results in [15].
middle column, thus effectively using 6 cells in total instead of all 18 cells as used in the case of self capacitance. Due to fewer readings used by interpolation algorithm, the true location estimated by a 3 point interpolator may not be as accurate as in the case of self capacitance. Note that mutual capacitance calls for a separate pin-out dedicated to every single row and column in the screen. In the case of large touch screens (such as kiosks), the increasing number of pin outs can raise costs and increase the area occupied by the sensing circuit. Also it is not possible to design larger size grid elements to reduce pin outs, because that will result in the loss of precision. Consequently, the transformation from a small-scale design to a large scale one is not a simple process. It involves study of cost and manufacturability, and a trade-off has to be established between feasibility and accuracy while designing sensor patterns. The first step in obtaining an optimal grid pattern which can be used for both small and large scale touch panels without extra cost and performance degradation is to define a methodology to test accuracy of existing grid patterns. In this paper we have devised such a methodology for different grid pattern designs.

The simulations were performed with and without noise and the results are shown in Table II and Figure 6. As explained in Section II, there are numerous grid patterns proposed by designers over the years. It is not feasible to design, fabricate and test these grid patterns in hardware because of the resources required (accurate motion stage apparatus) and time needed. A simulation environment, on the other hand provides a viable option to construct a methodology for these grid patterns as it is easy to simulate a complex design (for example hybrid pattern, shown in Figure 13) and environmental variables can be controlled to test under different conditions.

As it is evident from the simulation results that although interleaved pattern performs better than the diamond and angled cross pattern for all the four estimators, the best result is from the Gaussian interpolation algorithm (see Table I, II and III). But as it is still not visible why the interleaved performs better, we used Fourier analysis of all the panels in order to find out that whether the magnitude plays an important role in defining the performance of the pattern or the phase. As this paper treats the grid patterns of touch panels as images, Fourier analysis may prove useful in studying their performance.

Two dimensional fast Fourier transform was applied on the basic elements of the grid patterns. Log magnitude and phase plot of the results is shown in Figure 12.

Fig. 12: Fourier log magnitude and phase for grid patterns. It can be seen from the magnitude plot that both angled cross and diamond pattern have high frequency values at multiple of 45 degrees whereas the interleaved pattern has high frequency values at multiples of 90 degrees.

In order to find out that what component of the Fourier (magnitude or phase) contributes more to the performance of the grid patterns, a new grid pattern was devised, compromising of magnitude from the diamond pattern and phase from the interleaved pattern. As phase determines where the frequency components are situated in an image [18] thus in theory the hybrid pattern performance curve should lie in between the diamond and the interleaved pattern. The basic element and the touch panel of the hybrid grid is shown in Figure 13.

All the three patterns were passed through the angle sweep movement test in order to get a reading on all possible angles from 0 to 360 degrees. The simulation results are shown in Table IV.

Fig. 13: (a) Hybrid pattern basic element (b) Hybrid panel made by repeating the basic element in rows and columns.

It is clear from Table IV that hybrid pattern performance for COM, Gaussian and Parabolic estimators is in between
Fig. 14: Fourier unwrapped phase over a line for all three panels. The first, second and third column indicate the phase response of diamond, interleaved and angled cross respectively. The first row shows the phase over the middle row and the second row shows the phase over the middle column in a basic element of the grid pattern.

TABLE IV: Performance comparison of hybrid pattern with diamond and interleaved pattern. The hybrid pattern performance lies in the middle of diamond and interleaved pattern.

<table>
<thead>
<tr>
<th>Angle Sweep</th>
<th>Diamond Grid Mean Error (mm)</th>
<th>Interleaved Grid Mean Error (mm)</th>
<th>Hybrid Grid Mean Error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM</td>
<td>0.106</td>
<td>0.084</td>
<td>0.089</td>
</tr>
<tr>
<td>Linear</td>
<td>0.252</td>
<td>0.251</td>
<td>0.248</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.044</td>
<td>0.035</td>
<td>0.050</td>
</tr>
<tr>
<td>Parabolic</td>
<td>0.203</td>
<td>0.195</td>
<td>0.196</td>
</tr>
</tbody>
</table>

diamond and interleaved (indicated by bold text in Table IV) thus supporting the hypothesis that phase is the major player in determining the performance of a grid pattern.

Unwrapped phase of Fourier transform over a line in the basic element of all three patterns (diamond, interleaved and angled cross) was also studied. The plots in Figure 14 show that the interleaved pattern has a highly non-linear phase but the angled cross and diamond pattern show linear tendencies in their phase responses. This observation shows an interesting phenomenon that the linearity and non-linearity of the phase affects the performance of grid design.

VI. Summary

This paper discusses the most common method of projected capacitive sensing, that of absolute capacitance sensing. To provide a universal methodology of comparing the performance of different capacitive touch screen grid patterns was the main objective of this paper. We compared the performance of three different kinds of grid designs using four interpolation algorithms. The methodology presented in this paper can be extended to any sensor pattern design. Another major conclusion came from the comparison between the interpolation algorithms that the Gaussian algorithm performed better than the others by a large margin, thus supporting the results in [15]. How Fourier transform and non-linearity of phase can be used to construct an optimal pattern will be investigated further in the future. The results shown in this paper are based on the fact that both finger and noise model is Gaussian. A more realistic finger and noise model along with hardware testing using a robotic finger will be applied in the future to confirm our results and to find out what makes an optimal grid pattern for capacitive touch screens.

ACKNOWLEDGEMENT

We would like to thank Prof. Zachi Baharav of Cogswell Polytechnical College for his helpful suggestions and ideas.

REFERENCES

Humza Akhtar is currently working as an Research Associate in Earth Observatory of Singapore. He is also a Ph.D. student pursuing his Ph.D. in Capacitive touch screen grid patterns from School of Computer Engineering, Nanyang Technological University under the supervision of Prof. Ramakrishna Kakarala. Prior to joining Earth Observatory of Singapore, he obtained his Masters degree from Nanyang Technological University in 2011 with specialization in Embedded Systems. His research interests include capacitive grid patterns of touch screens, field programmable gate arrays and unmanned aerial vehicles. Contact email: hakhtar@ntu.edu.sg

Ramakrishna Kakarala (SM ’01) is an Associate Professor in Computer Engineering at the Nanyang Technological University (NTU) in Singapore, a position he has held since 2008. Prior to joining NTU, he was a Master Scientist with Avago Technologies in San Jose, California, and a Research Engineer with Agilent Laboratories. He has also worked as a Lecturer (promoted to Senior Lecturer) at the University of Auckland, New Zealand, from 1992-1999. He received the Ph.D. in Mathematics from the University of California, Irvine, where he was awarded a Chancellor’s Fellowship, and the M.S.E.E. and B.S.Comp.E. from the University of Michigan, Ann Arbor.