Keywords: touch screen panels, touch screen controllers, touch screen, touch panels, N-point calibration, calibration, 3-point calibration, touch systems, resistive
Related Parts

An Easy-to-Understand Explanation of Calibration in Touch-Screen Systems

By: Faisal Tariq

Abstract: Mechanical misalignment and scaling factors lead to a mismatch between the values coming from a touch screen panel (as translated by a touch screen controller) and the display (typically an LCD) on which the touch screen panel is mounted. This tutorial discusses how to calibrate the touch screen panel to match the display.

A similar version of this article appeared in the April 20, 2012 issue of *EE Times* magazine.

Introduction

Mechanical misalignment and scaling factors can lead to a mismatch between the values from a touch-panel system (in this tutorial, the touch-panel system implies a setup comprised of a touch screen and a touch-screen controller) and the display (typically an LCD) on which the touch screen is mounted. This article discusses the mathematical techniques to calibrate the touch-panel system so you can match the graphics on the display to the output from the touch-screen controller.

Understanding Image Transformation: Translation, Rotation, and Scaling

Figure 1 is an exaggerated view of the distortion that might happen to a circle being displayed on an LCD touch-screen display. When a finger is traced around the circle (red line), the touch-panel system may give out the coordinates of an ellipse (blue line) instead of the circle, shown below in Figure 1. This change of the shape from a circle to an ellipse can be explained by the following graphic transformations: translation, rotation, and scaling.
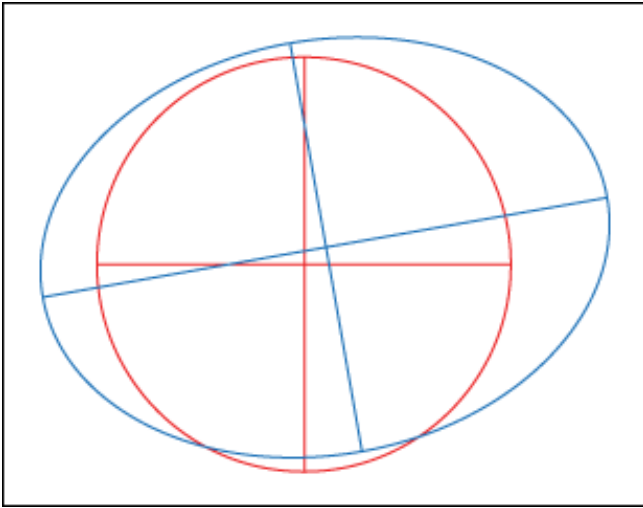
Figure 1. A circle changes shape on a touch screen because of a mismatch between the display and the touch-panel system.

Intuition suggests that any coordinate point x, y in an x-y plane that has undergone a transformation should look like:

x_{NEW} = f1(x_{OLD}, y_{OLD}) + constant1   (Eq. 1a)
y_{NEW} = f2(x_{OLD}, y_{OLD}) + constant2   (Eq. 1b)
Where x_{NEW} and y_{NEW} are the transformed coordinates; x_{OLD} and y_{OLD} are the old coordinates; f1() and f2() are functions that transform the old coordinates; constants1 and 2 are just that, constants.

If the transformation is linear, then functions f1() and f2() can be replaced by the following equations:

x_{NEW} = A x_{OLD} + B y_{OLD} + C   (Eq. 2a)
y_{NEW} = D x_{OLD} + E y_{OLD} + F   (Eq. 2b)
Where A, B, C, D, E, and F are constant coefficients.

Note that f1() = A x_{OLD} + B y_{OLD} and f2() = D x_{OLD} + E y_{OLD}, where constant1 and constant2 are C and F, respectively.

The Mathematics Behind Translation, Rotation, and Scaling
An example, showing a graphic transformation, will demonstrate that the final transformation equations from this exercise will lead to the above equations 2a and 2b and, therefore, our intuition holds true. (The user can bypass this example and jump directly to the **Three-Point Calibration** and **N-Point Calibration** discussions below.)

The illustration below (**Figure 2**) shows a square (Shape A) that has undergone translation, rotation, and scaling (Shape B). In the process it is transformed into a rectangle.
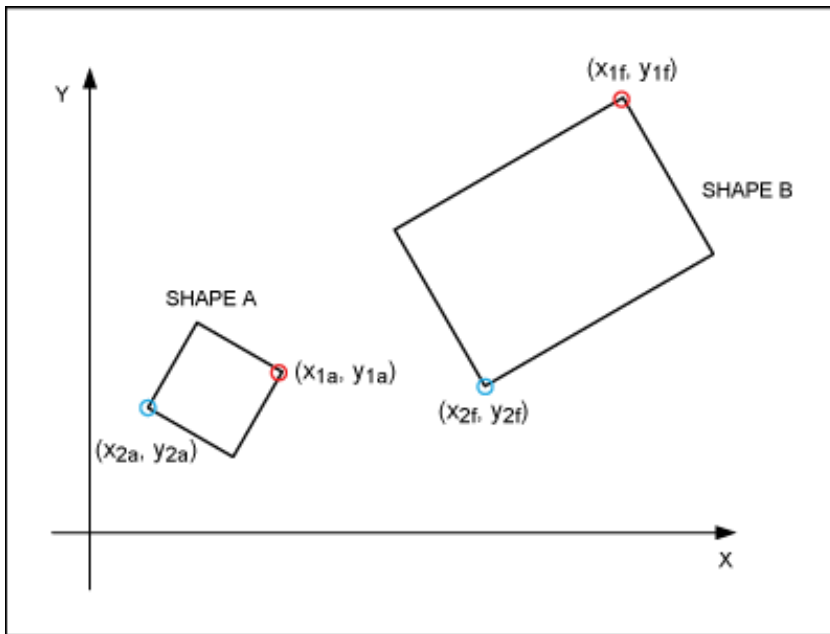
Figure 2. Translation, rotation, and scaling transform a square into a rectangle.

To rotate and scale the square (Shape A), it is important to move its center to the origin first (Figure 3) to ensure that the rotation and scaling happen uniformly.
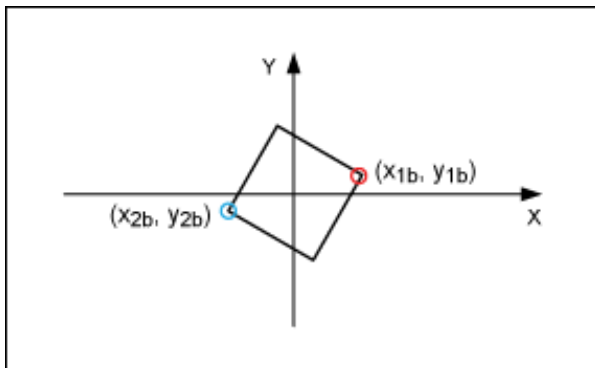

Figure 3. The square of Figure 2 (Shape A) is moved so that its center lies on the origin of the x-y axis.

This move to the center, or translation to the center, changes the value of all the coordinates of the square. For simplicity here, only the transformation of x_1, y_1 in Figure 2 will be explored. Consequently, x_{1a}, y_{1a} translates to x_{1b}, y_{1b} and the new equation representing this new figure is:

$$x_{1b} = x_{1a} + T_{x0} \tag{Eq. 3a}$$
$$y_{1b} = y_{1a} + T_{y0} \tag{Eq. 3b}$$

Rotate the above figure so that proper scaling can be done in the x and y directions. **Figure 4** shows that the figure has been rotated counterclockwise by angle θ.
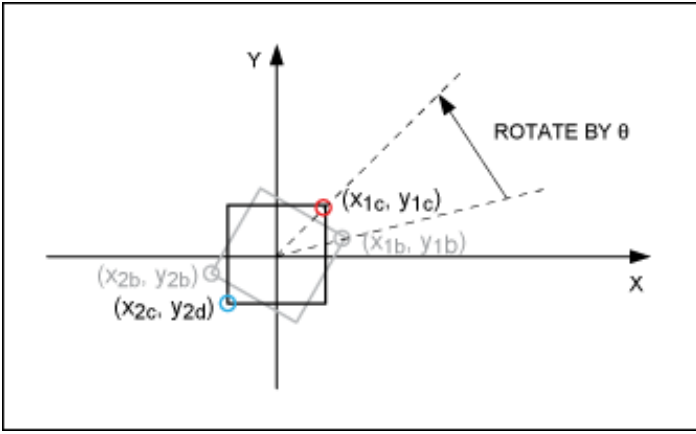
Figure 4. Following Equation 3, the square is rotated counterclockwise.

The new coordinates (x_{1c}, y_{1c}) after the rotation are shown below. (Please see the Appendix for the derivation of the equation for this rotation.)

$x_{1c} = x_{1b} \cos\theta - y_{1b} \sin\theta$ $\qquad\qquad$ (Eq. 4a)
$y_{1c} = x_{1b} \sin\theta + y_{1b} \cos\theta$ $\qquad\qquad$ (Eq. 4b)

Substitute Equations 3a and 3b into Equations 4a and 4b, and simplify:

$x_{1c} = x_{1a} \cos\theta - y_{1a} \sin\theta + K_x$ $\qquad\qquad$ (Eq. 5a)
$y_{1c} = x_{1a} \sin\theta + y_{1a} \cos\theta + K_y$ $\qquad\qquad$ (Eq. 5b)

Where $K_x = T_{x0} \cos\theta - T_{y0} \sin\theta$ and $K_y = T_{x0} \sin\theta + T_{y0} \cos\theta$.

Now scale the square in both the x and y directions to make it the same size as the rectangle in Figure 2 (Shape B). If we say that G_x is the scaling factor in the x direction and G_y is the scaling factor in the y direction, then the square will transform into a rectangle (**Figure 5**).
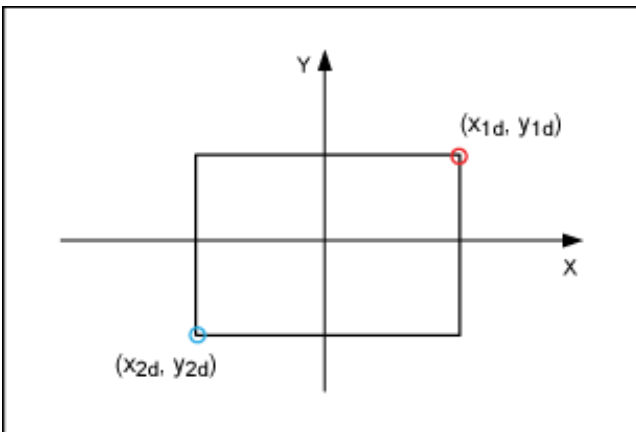

Figure 5. Transform the square into a rectangle using scaling factors G_x for the x direction, and G_y for the y direction.

Multiplying Equations 5a and 5b with the scaling factors G_x and G_y yields Equations 6a and 6b, which are the coordinates for the rectangle shown in Figure 5.

$x_{1d} = x_{1a} G_x \cos\theta - y_{1a} G_x \sin\theta + K_x G_x$ $\qquad\qquad$ (Eq. 6a)
$y_{1d} = x_{1a} G_y \sin\theta + y_{1a} G_y \cos\theta + K_y G_y$ $\qquad\qquad$ (Eq. 6b)

Rotate the rectangle by a so that it matches the rotational orientation of Shape B in Figure 2. This leads to the transformation of (x_{1d}, y_{1d}) into (x_{1e}, y_{1e}), as shown in the **Figure 6**.
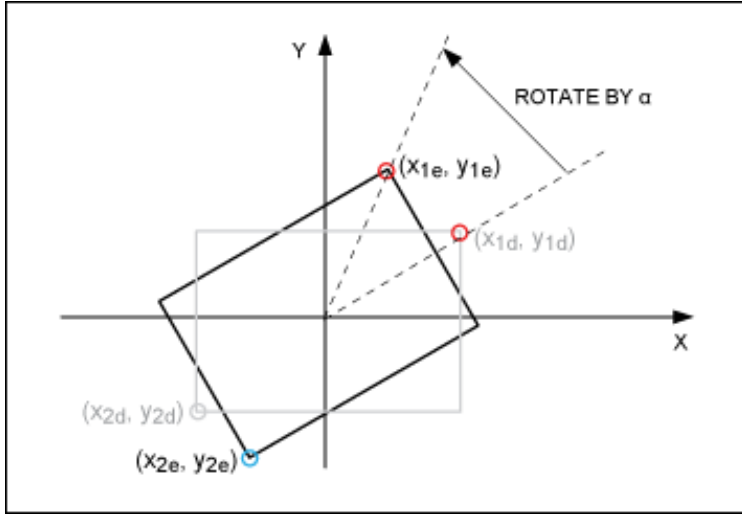

Figure 6. The rectangle of Figure 5 is rotated so that it matches the orientation of Shape B in Figure 2.

Once again the rotation is counterclockwise. The equations for (x_{1e}, y_{1e}) in terms of (x_{1d}, y_{1d}) are:

$x_{1e} = x_{1d} \cos\alpha - y_{1d} \sin\alpha + K_x$ (Eq. 7a)
$y_{1e} = x_{1d} \sin\alpha + y_{1d} \cos\alpha + K_y$ (Eq. 7b)

Substitute Equations 6a and 6b into Equations 7a and 7b, and simplify:

$x_{1e} = x_{1a} A + y_{1a} B + P$ (Eq. 8a)
$y_{1e} = x_{1a} D + y_{1a} E + Q$ (Eq. 8b)

Where:

$A = G_x \cos\theta \cos\alpha - G_y \sin\theta \cos\alpha$
$B = - G_x \sin\theta \cos\alpha - G_y \cos\theta \cos\alpha$
$P = K_x G_x \cos\alpha - K_y G_y \sin\alpha$
$D = G_x \cos\theta \sin\alpha + G_y \sin\theta \cos\alpha$
$E = -G_x \sin\theta \sin\alpha + G_y \cos\theta \cos\alpha$
$Q = K_x G_x \sin\alpha + K_y G_y \cos\alpha$

Now a translation (**Figure 7**) needs to be done to move the rotated rectangle to where Shape B appears in Figure 2.
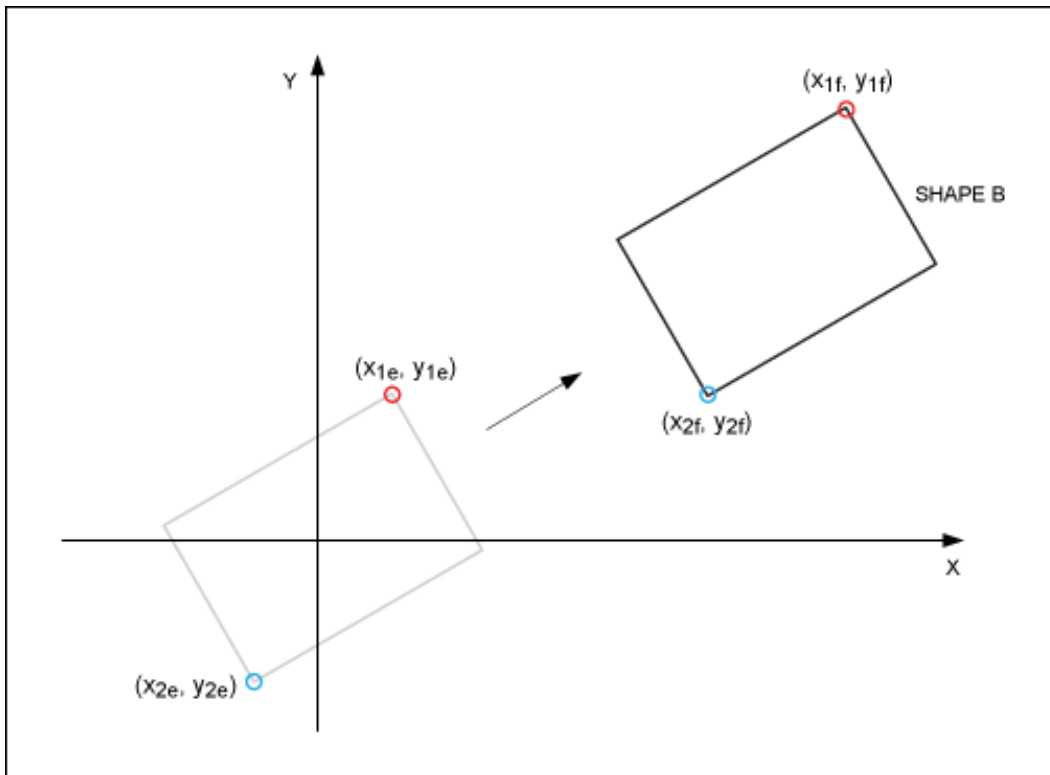
Figure 7. Translation of the rectangle from the origin to the location where Shape B appears in Figure 2.

Assume that the translation in the x and y direction is T_x and T_y, respectively. Therefore, Equations 8a and 8b will change to:

$$x_{1f} = x_{1a} A + y_{1a} B + C \hspace{6cm} \text{(Eq. 9a)}$$
$$y_{1f} = x_{1a} D + y_{1a} E + F \hspace{6cm} \text{(Eq. 9b)}$$

Where $C = P + T_x$ and $F = Q + T_y$.

Equations 9a and 9b are what we want—they match Equations 1a and 1b. Note that the coordinate (x_{1f}, y_{1f}) is in terms of (x_{1a}, y_{1a}). In the touch panel mounted on a display the user will have to determine the A, B, C, D, E, and F during calibration.

Please note that the purpose of this exercise is to give the user an appreciation of the general form of the equation for the transformation, shown by Equations 9a and 9b. This article shows that we will arrive at this general form irrespective of the direction and magnitude of rotation, translation, and scaling.

Three-Point Calibration
We start with a pair of equations, the final pair in our mathematical exercise above.

$$x_{1f} = x_{1a} A + y_{1a} B + C \hspace{6cm} \text{(Eq. 9a)}$$
$$y_{1f} = x_{1a} D + y_{1a} E + F \hspace{6cm} \text{(Eq. 9b)}$$

The goal of the touch-panel system calibration is to solve the Equations 2a and 2b (or similarly Equations 9a and 9b) to derive values for A, B, C, D, E, and F.

Looking at these equations we know that there are six unknowns. Therefore, we will need six equations to solve for these unknowns and this can be achieved by doing a three-point calibration for a touch-panel system. The user will generate three pairs of (x, y) coordinates by touching the panel at the three pairs of display coordinates: (x_{1d}, y_{1d}), (x_{2d}, y_{2d}) and (x_{3d}, y_{3d}). If their corresponding touch-screen values (as presented by the touch-screen controller) are (x_1, y_1), (x_2, y_2), and (x_3, y_3), then the six unknowns can be solved by the equations shown below. These points must be independent of each other, as shown in **Figure 8**.
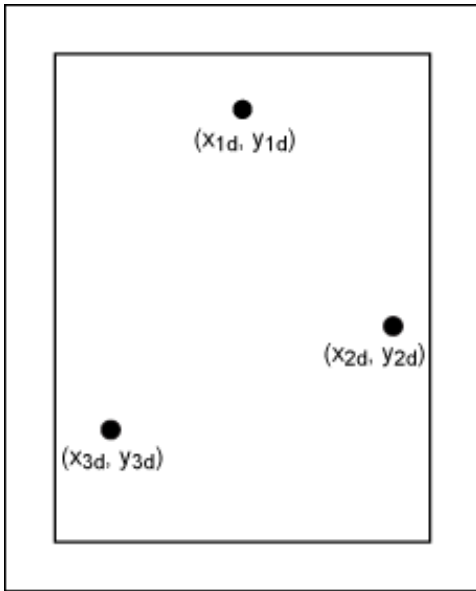

Figure 8. LCD panel showing three display coordinates for the user to touch.

$$x_{1d} = x_1 A + y_1 B + C$$
$$x_{2d} = x_2 A + y_2 B + C$$
$$x_{3d} = x_3 A + y_3 B + C$$

$$y_{1d} = x_1 D + y_1 E + F$$
$$y_{2d} = x_2 D + y_2 E + F$$
$$y_{3d} = x_3 D + y_3 E + F$$

Now with six equations for the six unknowns, the above can be written in matrix form as:

$$\begin{bmatrix} x_{1d} \\ x_{2d} \\ x_{3d} \end{bmatrix} = Z \times \begin{bmatrix} A \\ B \\ C \end{bmatrix} \qquad \begin{bmatrix} y_{1d} \\ y_{2d} \\ y_{3d} \end{bmatrix} = Z \times \begin{bmatrix} D \\ E \\ F \end{bmatrix} \qquad \text{Where, } Z = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}$$

Therefore, a little matrix manipulation will yield A, B, C, D, E, and F, as shown below:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = Z^{-1} \times \begin{bmatrix} x_{1d} \\ x_{2d} \\ x_{3d} \end{bmatrix} \quad \text{Eq. 10a} \qquad \begin{bmatrix} D \\ E \\ F \end{bmatrix} = Z^{-1} \times \begin{bmatrix} y_{1d} \\ y_{2d} \\ y_{3d} \end{bmatrix} \quad \text{Eq. 10b}$$

Where, Z^{-1} is the inverse matrix of Z.

Three-Point Calibration Example

We will use the MAX11800 touch-screen controller in this example. Assume that an LCD panel display has a resolution of 256 x 768 and that the three calibration points chosen are (65, 350), (200, 195), and (195, 550). These are the exact points that we want the touch panel to show when it is touched. However, since the resolution for the MAX11800 is 4096 x 4096 (12 bits) and because of mechanical misalignment, the coordinate will come out differently. For this exercise, we assume that these values are: (650, 2000), (2800, 1350), and (2640, 3500), respectively. *Please note: these values are made up and serve as an example only*.

Solving for A, B, C, D, E, and F using Equations 10a and 10b, we get the following:

A = 0.0635
B = 0.0024
C = 18.9116
D = -0.0227
E = 0.1634
F = 37.8887

Therefore, the equations that will generate the x and y coordinates for this particular example are:

x_d = 0.0635 x + 0.0024 y + 18.9116
y_d = -0.0227 x + 0.1634 y + 37.8887
Where (x, y) are the coordinates coming from the touch-panel system and (x_d, y_d) is the adjusted value.

N-Point Calibration

The user may choose to calibrate using more points. A generalized form of the sets of equations will look like the following:

$x_{1d} = x_1 A + y_1 B + C$
$x_{2d} = x_2 A + y_2 B + C$
$x_{3d} = x_3 A + y_3 B + C$ (Eq. set 11a)
...
$x_{nd} = x_n A + y_n B + C$

$$y_{1d} = x_1 D + y_1 E + F$$
$$y_{2d} = x_2 D + y_2 E + F$$
$$y_{3d} = x_3 D + y_3 E + F$$
... (Eq. set 11b)
$$y_{nd} = y_n D + y_3 E + F$$

Where, $(x_{1d}, y_{1d})...(x_{nd}, y_{nd})$ are coordinates generated by the display; $(x_1, y_1)...(x_n, y_n)$ are the corresponding values (for n points) determined by the MAX11800 from the touch-screen panel. The goal is to determine the coefficients using these values.

In the Equation set 11a there are three unknowns, A, B, and C, but the set of coordinates is more than three. This implies that there are more equations than unknowns. Therefore, in this instance it makes sense to use the least square fit to utilize all the points and derive an average value of the coefficients. This also means that more calibration points would help lower errors. The least square fit is represented in **Figure 9**. The same technique will be applied for determining the unknowns D, E, and F for the y values.
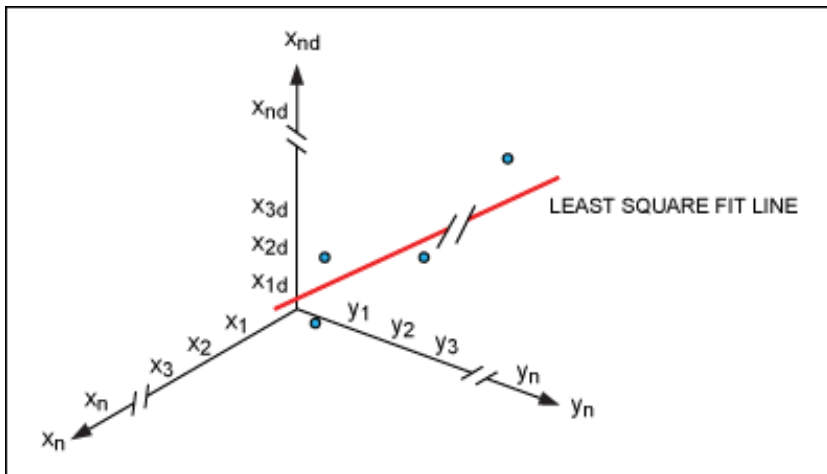

Figure 9. An average value for the coefficients A, B, and C is found by applying the least square fit to the points.

Equation sets 11a and 11b can be written in matrix form, as shown:

By using the least square fit in this matrix form, the coefficients are given by the equations below. (Please refer to a book on regression analysis for the derivation of the matrix form of the least square fit form.)

$$
\begin{bmatrix} A \\ B \\ C \end{bmatrix} = (Z^T \times Z)^{-1} \times Z^T \begin{bmatrix} x_{1d} \\ x_{2d} \\ x_{3d} \\ \dots \\ x_{nd} \end{bmatrix} \quad \text{Eq. 12a} \qquad \begin{bmatrix} D \\ E \\ F \end{bmatrix} = (Z^T \times Z)^{-1} \times Z^T \begin{bmatrix} y_{1d} \\ y_{2d} \\ y_{3d} \\ \dots \\ y_{nd} \end{bmatrix} \quad \text{Eq. 12b}
$$

N-Point Calibration Example

Once again, assume that an LCD panel display has a resolution of 256 x 768 and the five calibration points are (100, 350), (50, 200), (200, 200), (210, 600), and (65, 600). These are the exact points that we want the touch panel to show when it is touched. However, since the resolution for the MAX11800 is 4096 x 4096 (12 bits) and because of mechanical misalignment, the points may come out differently. Again assume that they are (1700, 2250), (750, 1200), (3000, 1500), (2500, 3400), and (600, 3000), respectively. *Please note: these values are made up and meant to serve as an example only*.

Solving for A, B, C, D, E, and F using Equations 12a and 12b, we get:

A = 0.0677
B = 0.0190
C = -33.7973
D = -0.0347
E = 0.2100
F = -27.4030

Therefore, the equations to generate the x and y coordinates for this particular example are:

x_d = 0.0677 x + 0.0190 y - 33.7973
y_d = -0.0347 x + 0.2100 y - 27.4030
Where (x, y) are the coordinates coming from the touch-panel controller, and (x_d, y_d) is the adjusted coordinate value to match what is shown on the display.

Example Summary

If the least squares technique is used on three points, it will yield the same coefficient as the three-point calibration. It is mathematically easier, therefore, to handle the three-point calibration as three simultaneous linear equations with three unknowns instead of the lengthier calculation with the least squares technique.

Appendix: Derivation of the Equation for a Counterclockwise Rotation
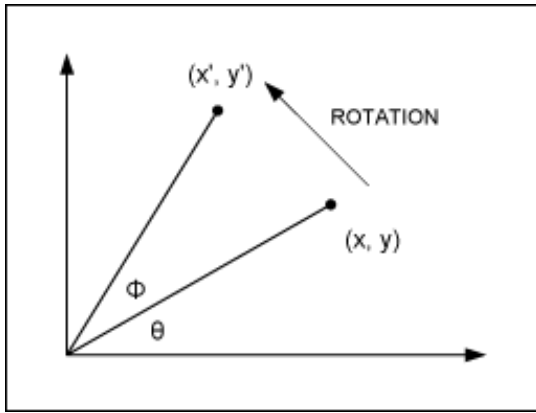
The original point (x, y) can be written in polar form as:

x = r Cosθ and y = r Sinθ, (Eq. A)
Where r is the radial distance from the origin to the (x, y) coordinate.

After the rotation the (x, y) coordinates become (x', y'):

x' = r Cos(θ + Φ) and y' = r Sin(θ + Φ) (Eq. B)
Using the trig identities below:

Cos(θ + Φ) = CosΦ Cosθ - SinΦ Sinθ
Sin(θ + Φ) = SinΦ Cosθ - CosΦ Sinθ (Eq. C)
And substituting in Equation B, we get:

x' = r CosΦ Cosθ - r SinΦ Sinθ
y' = r SinΦ Cosθ - r CosΦ Sinθ (Eq. D)
Substituting Equation A into Equation D:

x' = x CosΦ - y SinΦ
y' = x SinΦ - y CosΦ (Eq. E)
The derivation for the clockwise rotation is left as an exercise for the user.

Related Parts

MAX11800	Low-Power, Ultra-Small Resistive Touch-Screen Controllers with I²C/SPI Interface
MAX11801	Low-Power, Ultra-Small Resistive Touch-Screen Controllers with I²C/SPI Interface
MAX11802	Low-Power, Ultra-Small Resistive Touch-Screen Controllers with I²C/SPI Interface
MAX11803	Low-Power, Ultra-Small Resistive Touch-Screen Controllers with I²C/SPI Interface

MAX11811	TacTouch™, Low-Power, Ultra-Small, Resistive Touch-Screen Controller with Haptic Driver

Next Steps

EE-Mail	Subscribe to EE-Mail and receive automatic notice of new documents in your areas of interest.
Download	Download, PDF Format (247.4kB)
Share	g+1 Tweet Beğen ‹ 1 Other Channels Email this page to an associate or friend.

APP 5296: May 07, 2012
APPLICATION NOTE 5296, AN5296, AN 5296, APP5296, Appnote5296, Appnote 5296