

Reflection through an Arbitrary Line in a Mathematical Cartesian Coordinates

Hieu Vu

American University of Nigeria
98 Lamido Zubairu Way
Yola by Pass, P.M.B. 2250,
Yola, Adamawa State, Nigeria
Email: hieuvu@aun.edu.ng

ABSTRACT

Affine transformation preserves the original shape of an object therefore it is a very important aspect in computer graphics. This paper presents a technique, how to apply reflection transformation through an arbitrary line ($y = ax + b$) in a conventional mathematical Cartesian coordinates system.

Keywords: affine transformation, distortion, reflection, Cartesian coordinates system, arbitrary line, v-perp

1. INTRODUCTION

There are two types of transformations of objects. Distortion that can be achieved by moving the anchor points and control points, distortion transformations will change the original shape of objects. The other is affine transformation which preserves the fundamental shape of objects. The most important of affine transformations are: translation, scaling, rotation about a point, reflection about a line and shearing (distortion of an angle).

The universal tools for computer graphics programs are the matrices that manipulate points and vectors. Each point or vector on a plane is represented by a single column matrix[1].

This paper will present a direct method how to apply affine transformations: reflection through an arbitrary line ($y = ax + b$) in a mathematical Cartesian coordinates using analytic geometry.

2. PROBLEM

Reflections through the (x/y) axis and the origin are simple and easy to compute using direct calculation approach. Given point $P(x_p, y_p)$, we can find point $Q(x_q, y_q)$, the image of P through:

1. X-axis: $x_q = x_p$ and $y_q = -y_p \implies Q(x_q, y_q) = Q(x_p, -y_p)$

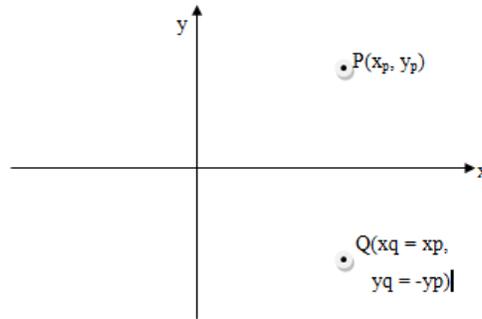


Figure 1: Reflection through x-axis

2. Y-axis: $x_q = -x_p$ and $y_q = y_p \implies Q(x_q, y_q) = Q(-x_p, y_p)$

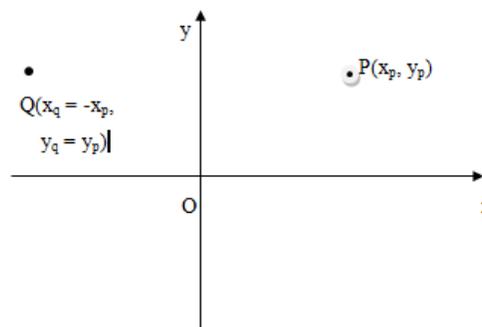


Figure 2: Reflection through y-axis

3. Origin O: $x_q = -x_p$ and $y_q = -y_p \implies Q(x_q, y_q) = Q(-x_p, -y_p)$

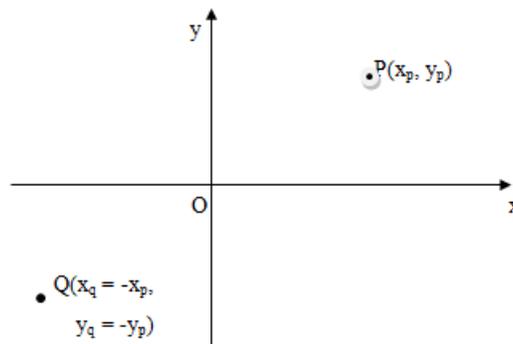


Figure 3: Reflection through the origin O

However, reflection through an arbitrary line $y = ax + b$ or any line on the Cartesian coordinate would require a lot more works either by transformation matrices manipulation or by direct computational approach.

3. REFLECTION THROUGH an ARBITRARYLINE: $y = ax + b$

Reflections are the simplest operation of transformations. Simple reflections will reflect the object through the x-axis, y-axis, and the origin. Reflection through an arbitrary line ($y = ax + b$) is much more complicated and requires some mathematical calculations.

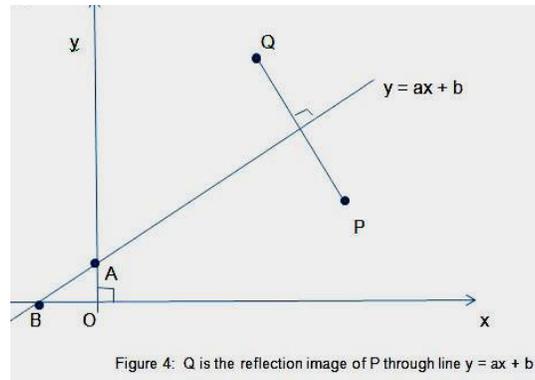


Figure 4: Q is the reflection image of P through $y = ax + b$

3.1. Algorithm Reflection about an Arbitrary Line Using Matrices

This can be done using a combination of other transformation operations: translate, rotate and reflect.

1. Translate the Cartesian coordinates so that the arbitrary line $y = ax + b$ will pass through the origin O.
 2. Rotate the Cartesian coordinates so that one of the coordinate (x-coordinate is preferable) lies onto the arbitrary line $y = ax + b$
 3. Perform the reflection about the aligned axis (also $y = ax + b$)
 4. Restore the original Cartesian coordinate by applying inverse rotation, and translation transformations.
- [2]

This algorithm is easy to understand, but implementation requires several matrices operations. This process is complicated and time consuming in term of executing efficiency. We also need to find the angle of rotation so the x-axis (or y-axis) to lie on the line $y = ax + b$ (step 2). And then after performs the reflection, we will have to find the inverse matrices to move the coordinates back to its original position (step 4).

3.2. Algorithm to Find Point Q Reflection of Point P through Line $y = ax + b$

This is a direct approach, using analytic geometry technique (Figure 4).

1. Find equation of line (PQ) that passes through point P and perpendicular to the line $y = ax + b$
2. Find the distance (d) from point P to the line $y = ax + b$.
3. Coordinates of point $Q(x_q, y_q)$ should satisfied equations of line (PQ) (from #1) and the distance $d_{(q)}$ is also equal the distance from $d_{(p)}$ or: $d_{(q)} = d_{(p)} = d$ (from #2)

3.2.1. Implementation Algorithm (3.2)

Problem: Find the image $Q(x_q, y_q)$ of point $P(x_p, y_p)$ that reflects through the line $y = ax + b$

Solution:

1. The line (PQ) perpendiculars to line $y = ax + b$ has slope $m = -1/a$ and goes through $P(x_p, y_p)$ must satisfy the equation:

$$y_p = (-1/a)x_p + b \quad (1)$$

Equation of line $y = ax + b$ and point $P(x_p, y_p)$ is given, substitute into equation (1) we have constant b.
 (Step 1) $b = y_p - (-1/a) x_p$

2. The line $y = ax + b$ will intersect x-axis and y-axis at A (x-intercept) and B (y-intercept). From the two points A and B, we can find vector v represents line $y = ax + b$: $v = (x_b - x_a, y_b - y_a)$. The vector that perpendiculars to vector v is called v^\perp (v-perp).

$$v^\perp = [-(y_b - y_a), x_b - x_a] = (y_a - y_b, x_b - x_a) \quad (2)$$

The distance d_p from point P to the line $y = ax + b$ is:

$$d_p = d = |v^\perp \cdot (x_p - x_a, y_p - y_a)| / |v^\perp| \quad (3) \quad [3]$$

Note: $(x_p - x_a, y_p - y_a)$ is vector (AP) that goes through points A and P.

Similarly, the distance d_q from Q to the line $y = ax + b$ is:

$$d_q = d = |v^\perp \cdot (x_q - x_a, y_q - y_a)| / |v^\perp| \quad (4)$$

3. From (steps 1 and 2) above, we have two equations with two unknowns (x_q, y_q) . Substitute y_q into equation (4), to solve for x_q , then y_q followed.

Example: Given point P(5, 2), find the image Q(x_q, y_q) of P that reflects through an arbitrary line $y = x + 1$

Solution

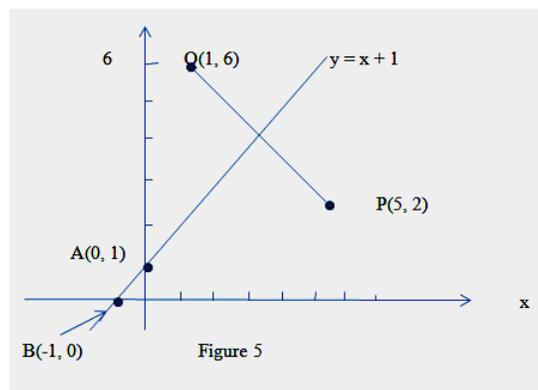


Figure5: Q is the reflection image of P through $y = ax + b$

Step 1: Line (PQ) is perpendicular to arbitrary line $y = x + 1$, therefore it has slope $m = -1 / 1 = -1$ and goes through point P, we have:

$$y_p = -x_p + b \quad \text{or} \quad 2 = -5 + b$$

$$\implies b = 7$$

$$\implies y_{(pq)} = -x_{(pq)} + 7 \quad (5)$$

Step 2: Find vector v^\perp that perpendiculars to vector v (line $y = x + 1$) using equation (2)

$$v^\perp = [-(y_b - y_a), x_b - x_a] = (y_a - y_b, x_b - x_a) = (-1, 1)$$

The distance $d_p = d$ is computed using equation (3)

$$\begin{aligned} d_p = d &= |v^\perp \cdot (x_p - x_b, y_p - y_b)| / |v^\perp| \\ &= |(-1, 1) \cdot (6, 2)| / \sqrt{(-1)^2 + (1)^2} = \\ 4 / \sqrt{2} &= 2\sqrt{2} \end{aligned} \quad (6)$$

The distance d_q from Q to the line PQ is also the same.

Step 3: Substitute $y_q = -x_q + 7$ (equation 5) into equation (6) with respect to point Q, we have:

$$d_q = d_p = \frac{|v^\perp \cdot (x_q - x_b, y_q - y_b)|}{|v^\perp|}$$

$$= |(-1, 1) \cdot [(x_q - (-1)), (-x_q + 7)]| / \sqrt{2}$$

$$= 2\sqrt{2}$$

$$= |(-1, 1) \cdot [(x_q + 1), (-x_q + 7)]| / \sqrt{2}$$

$$= 2\sqrt{2}$$

$$\implies |-x_q - 1 - x_q + 7| = 4$$

$$\text{or } |-2x_q + 6| = 4$$

$$\implies x_q = 1$$

Substitute x_q into equation (5), we have $y_q = 6$, so point Q(1, 6) is the image of point P(5, 2) that reflects through line $y = x + 1$

4. CONCLUSION

The transformations in graphics are linear transformations such as scaling, translation (reflection), shearing, rotation, and orthographic projection [4]. This paper can be used as a guide or teaching materials for computer graphics course. As can be seen through the example above, the reflection process of an object through an arbitrary line $y = ax + b$ in a conventional coordinate system can be solved by a direct approach method as in algorithm (3.2). For graphics programming, this method can be implemented in OpenGL, the official language for computer graphics or a high-level programming languages such as C++ or Java. In Java, affine transformations can be accomplished using class *java.awt.geom.AffineTransform* [5]. Other transformations such as scaling, rotation, translation, and shearing can be done in the same way.

References:

[1] Peter Shirley, Steve Marschner, "Fundamental of Computer Graphics" 3e, CRC Press, Boca Raton, Florida. 2009

[2]<http://nptel.ac.in/courses/Webcourse-contents/IITDelhi/Computer%20Aided%20Design%20&%20ManufacturingI/mod2/05.htm> 9/25/2014 12:06

[3] Hill, Kelley, "Computer Graphics Using OpenGL", (3e), Prentice Hall 2007, ISBN: 0-13-149670-0. Page 146.

[4] R. J. Renka, "OpenGL Transformation", Department of Computer Science & Engineering, University of North Texas, 2013

[5] David J. Eck, "Fundamental of Computer Graphics", Hobart and William Smith Colleges, 2010, page 12