3D Graphics on Atari 7800

Joey Fernau, Professor Shawn Blanton, and Professor Franz Franchetti

Carnegie Mellon Electrical & Computer Engineering

ACTL Carnegie Mellon Advanced Chip Test Laboratory

Abstract

The Atari 7800 is a 1986 video game console. It uses a custom 6502 processor called SALLY. It also uses a custom graphics chip called MARIA, which is unlike any other console graphics chip. 7800basic is a new language to program games for the Atari 7800. There are multiple limitations such as the 48KB ROM space on game cartridges, 4KB of RAM, and the unpolished 7800basic language. The goal is to use projective geometry to display 3D images using the Atari 7800 computer.

Projective Geometry

Projective geometry is a superset of standard Euclidean geometry. It is the study of properties that are invariant when projected. Given a camera position (x, y, z, theta, phi, psi), projective geometry can be applied to map 3D points in space to a 2D image. This creates the feeling of a 3D environment. See figure 1. Rotations, translations, and projections were used in this Python program. Equation 1 shows the formula used to convert a 3D point to the 2D screen the user views.

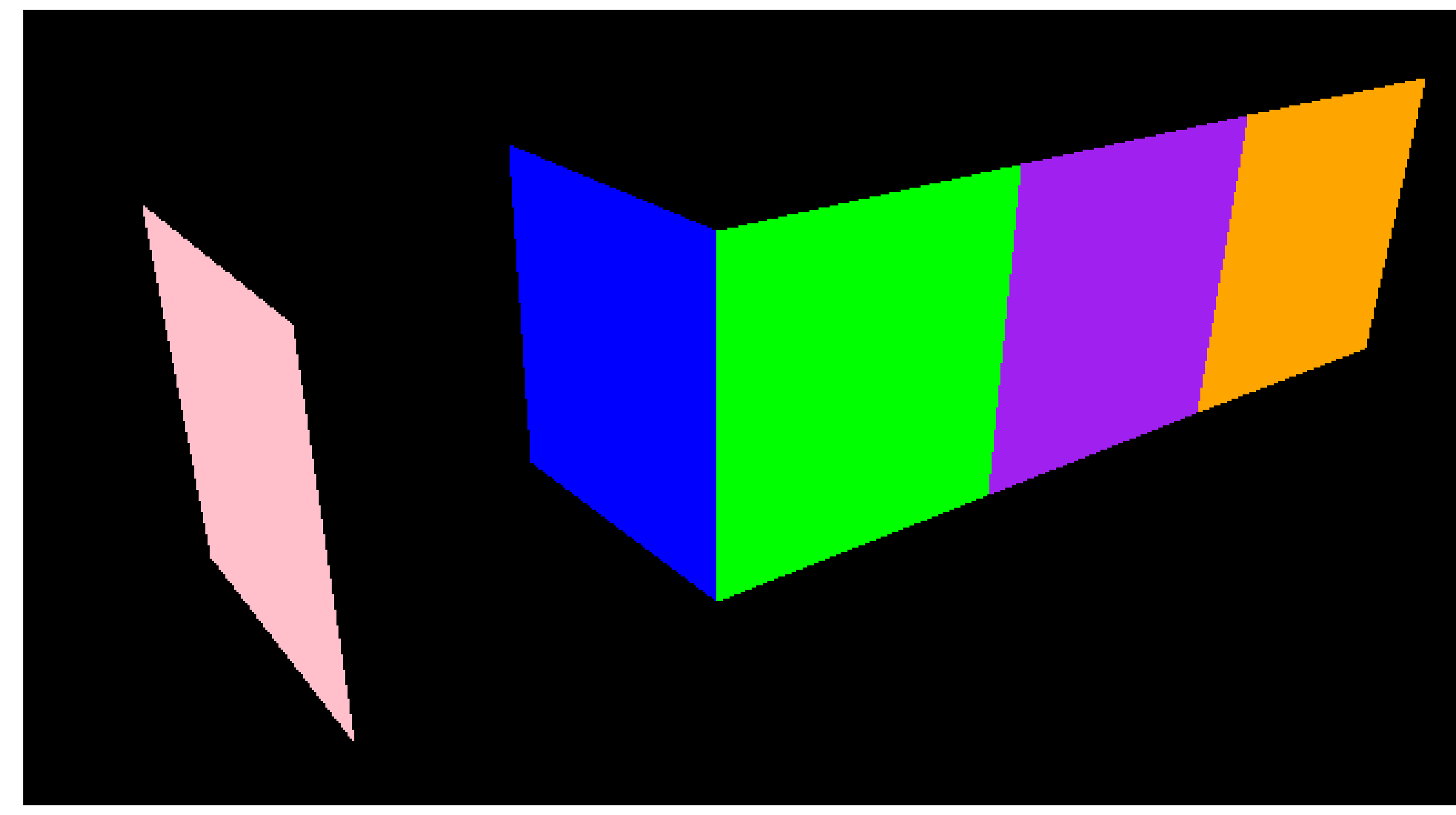

Figure 1: Written in Python using Tkinter. Using projective geometry to display “walls”. 3D points corresponding to a given wall’s vertices were mapped to this 2D image.

Fixed Point Numbers

7800basic has fixed point numbers, which contain the integer part of the number and the fractional part over 256. These fixed point numbers can only be used in addition and subtraction though. Thus, other methods of multiplication and division have to be used to perform these operations, such as repeated addition. Figure 3 displays the error in fixed point multiplication.

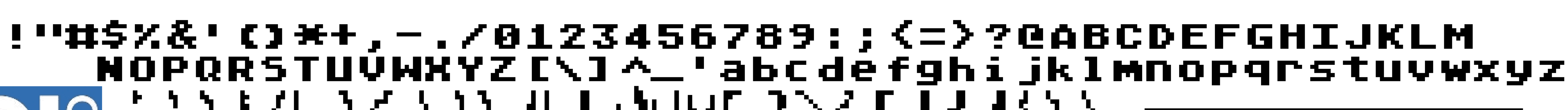

Figure 2: 8 by 8 sprites are used to display graphics. The bottommost row are some of the sprites used in generating the 3D to 2D projected image for the Atari.

Solution

7800basic displays images only through 8 by 8 pixel sprites. It also allows for simple animations of these sprites. After implementing trigonometry, fixed point multiplication and division, and matrices, these sprites are animated and translated across the screen to simulate a 3D environment.

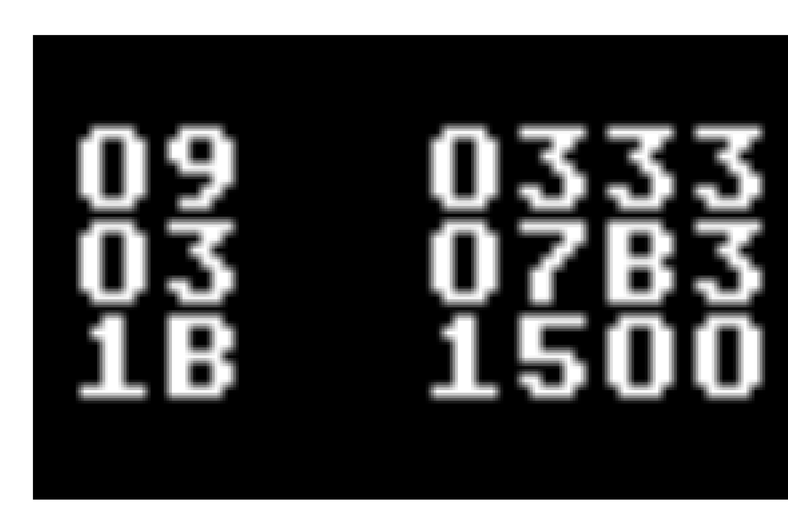

Figure 3: On left, multiplying 9 and 3 to produce 27. On right, multiplying (3 + 33/256) and (7 + 179/256) to produce (21 + 0/256), which is incorrect!

$$\underbrace{\begin{bmatrix} \frac{480}{2} & 0 & 0 & \frac{480}{2} \\ 0 & -\frac{480}{2} & 0 & \frac{480}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Screen}} \underbrace{\begin{bmatrix} \frac{480}{480\tan(60/2)} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(60/2)} & 0 & 0 \\ 0 & 0 & \frac{100+.1}{.1-100} & \frac{2(100)(.1)}{.1-100} \\ 0 & 0 & -1 & 1 \end{bmatrix}}_{\text{Projection}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) & 0 \\ 0 & \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-\phi) & 0 & \sin(-\phi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\phi) & 0 & \cos(-\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-\psi) & -\sin(-\psi) & 0 & 0 \\ \sin(-\psi) & \cos(-\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Rotation}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Translation}}$$

Equation 1: Conversion from a vector representing a point in 3D space to a vector representing a point on the 2D screen displayed to the user.

References

Atari Museum. http://www.atarimuseum.com/

An Introduction to Projective Geometry (for computer vision). http://robotics.stanford.edu/~birch/projective/projective.html