Air University Islamabad

FACULTY OF COMPUTING & ARTIFICAL INTELLIGENCE

Department of Creative Technologies

Project Report

Submitted To: Ma’am Nomana Abid

Submitted By: Malik Usama Arif (233105)
M. Abdullah Younas (232534)
Areeb Sehar (232559)
Huma Imtiaz (232556)
Marwa Shoukat (232547)

Submission Date: 8th June, 2025

Contents

Matrix Transformations in $\mathbb{R}^2$ Using Python

1. Introduction

Matrix transformations are a fundamental concept in linear algebra and computer graphics, enabling the manipulation of geometric figures in a coordinate space. This project explores two specific transformations.

Shear along the x-axis and compression along the x-axis and applies them to a 2D cartoon "Phineas".

2. Objective

- To apply a shear transformation with factor -3 along the x-axis.
- To apply a compression transformation with factor ½ along the x-axis.
- To implement these transformations using Python with NumPy and matplotlib.
- To visualize the transformation sequence on a custom-drawn 2D figure.

3. Mathematical Background

3.1 Shear Transformation (x-axis)

A shear along the x-axis modifies the x-coordinate based on the y-coordinate.

Transformation matrix:

$$\text{Shear}_x(k) = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$$

For k = -3:

$$\begin{bmatrix} 1 & -3 \\ 0 & 1 \end{bmatrix}$$

3.2 Compression Along x-axis

A compression by a factor ccc along x-axis multiplies x-coordinates by c.

Transformation matrix:

$$\text{Compress}_x(c) = \begin{bmatrix} c & 0 \\ 0 & 1 \end{bmatrix}$$

For c = ½:

$$\begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$$

4. Implementation

We used Python with the following libraries:

- NumPy: For matrix calculations.
- Matplotlib: For 2D plotting of shapes.

4.1 Original Figure Drawing

The initial cartoon figure was constructed using predefined coordinate points for the:

- Hair
- Face
- Eyes
- Lips

Each component was drawn using a polygon based on (x, y) points.

4.2 Shear Transformation

We applied the shear transformation to all points using matrix multiplication. The result skewed the shape horizontally, slanting it leftward due to the negative factor.

4.3 Compression Transformation

After shearing, we applied a compression transformation along the x-axis. This horizontally squeezed the already-sheared shape, giving the final output.

5. Code Summary

We created the following functions:

- draw_shape(): To render each shape on the coordinate system.
- shear_x(): Applies shear matrix to a list of (x, y) points.

- compress_x(): Applies compression matrix to a list of (x, y) points.

The transformation sequence was:

1. Original → Sheared
2. Sheared → Compressed
3. Original -> compressed

---

6. Results and Observations

- The original cartoon was clearly defined.
- After **shearing**, the shape was significantly slanted toward the left, particularly noticeable in features like the hair and face.
- After **compression**, the figure retained its slant but appeared “narrower” along the x-axis.

---

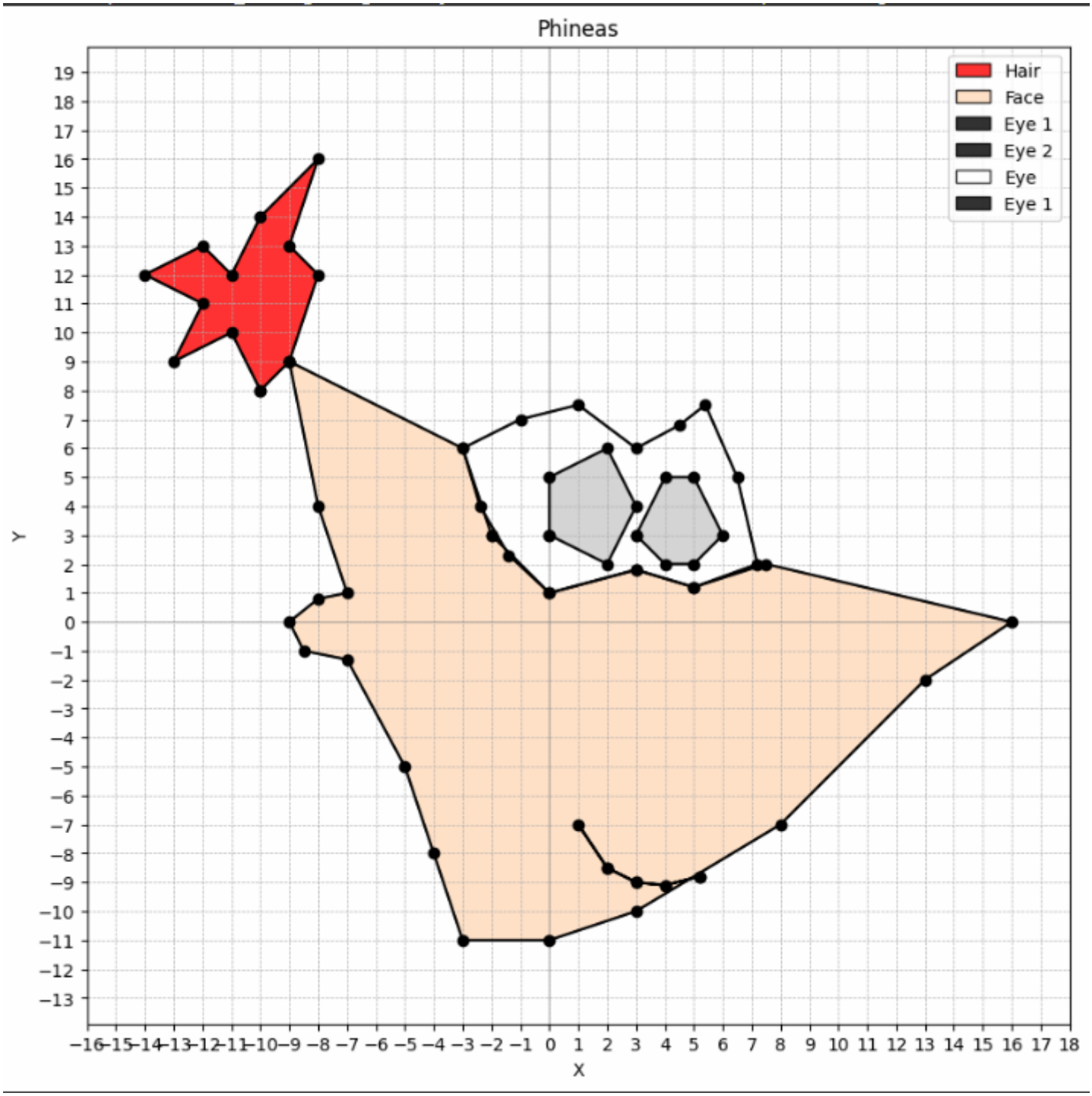
7. Conclusion

This project successfully demonstrated:

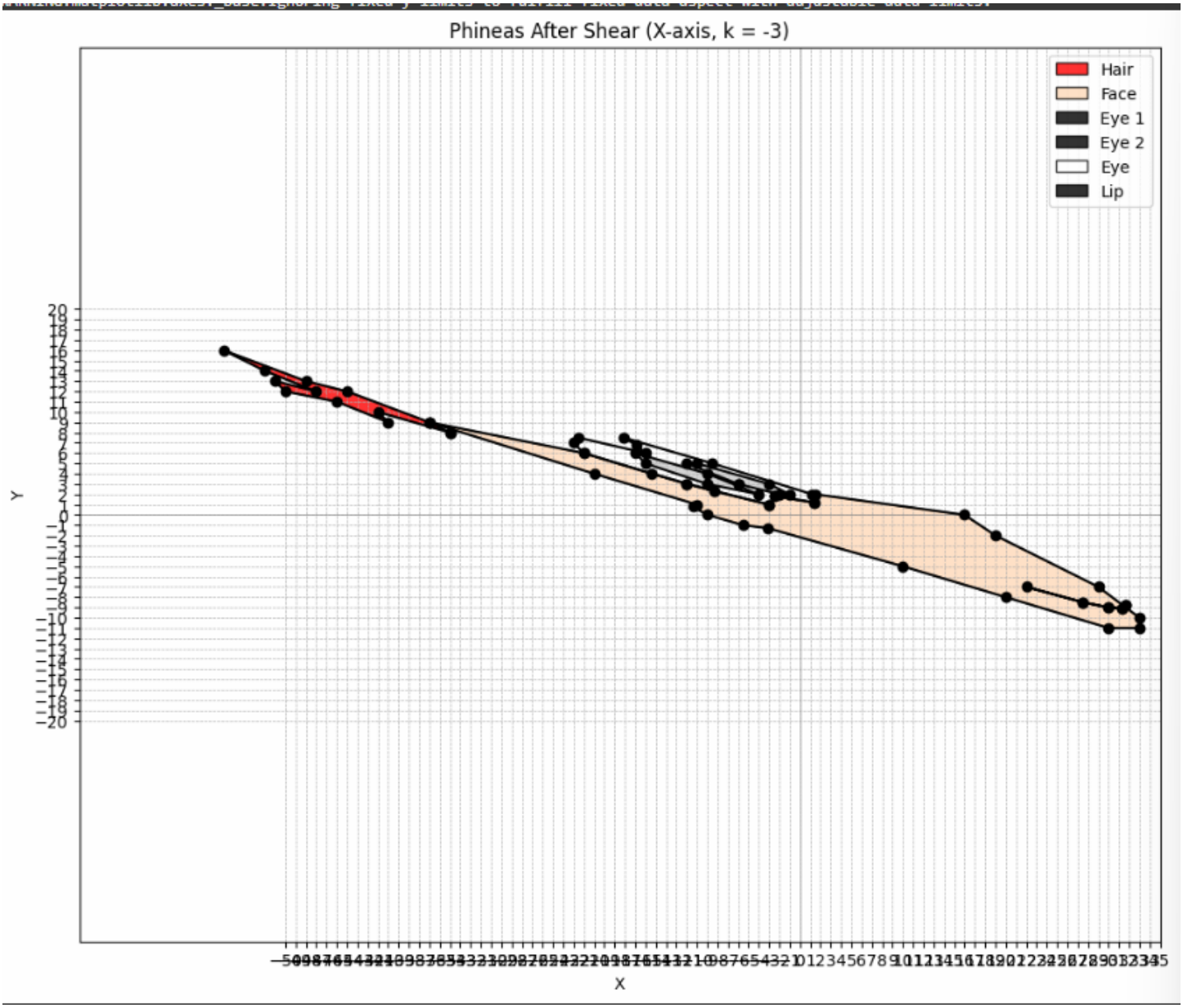
- The application of matrix transformations in 2D space using Python.
- The effects of shear and compression on geometric shapes.
- Visualization of composite transformations.

8. Screenshots

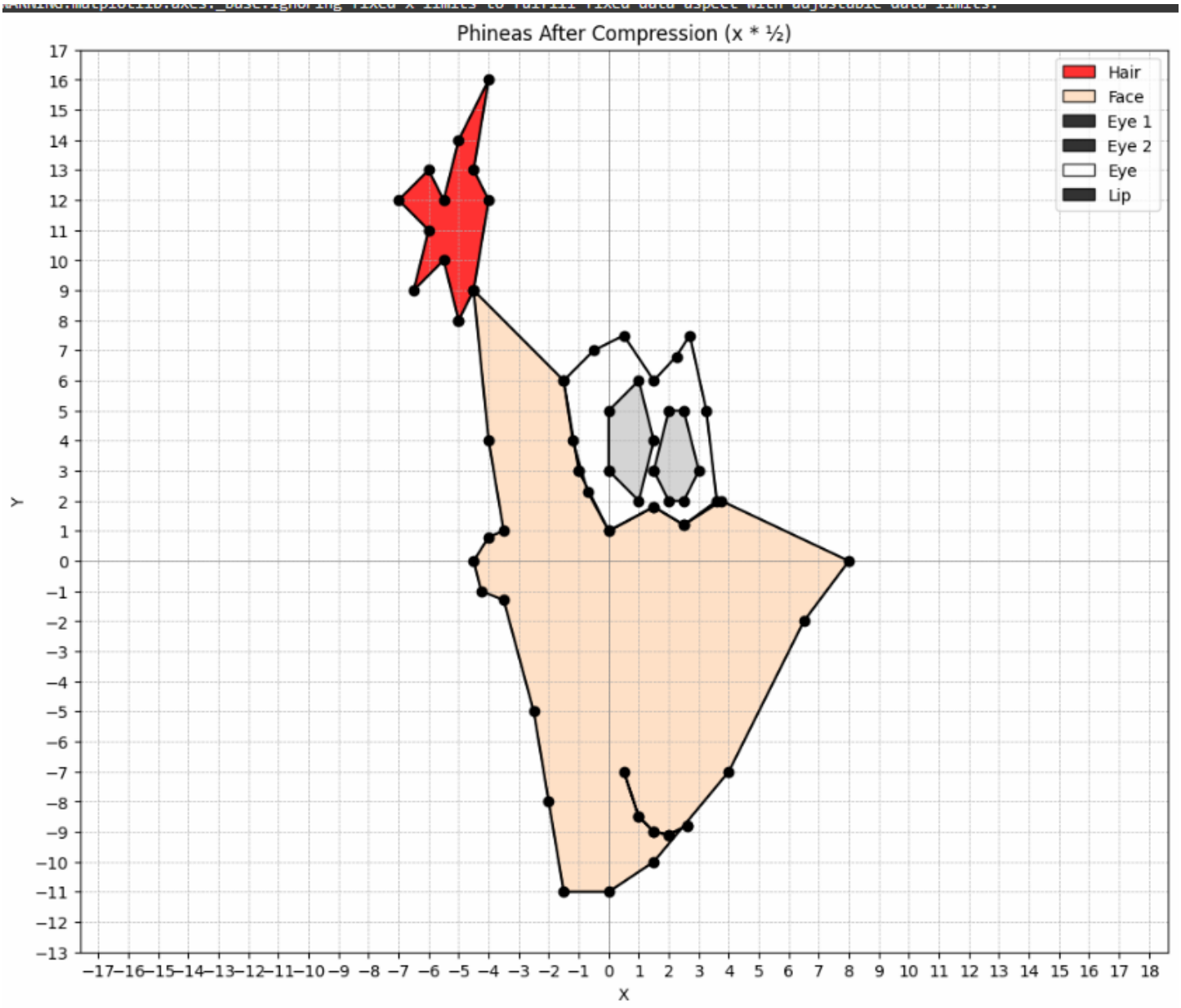
Original Cartoon Figure



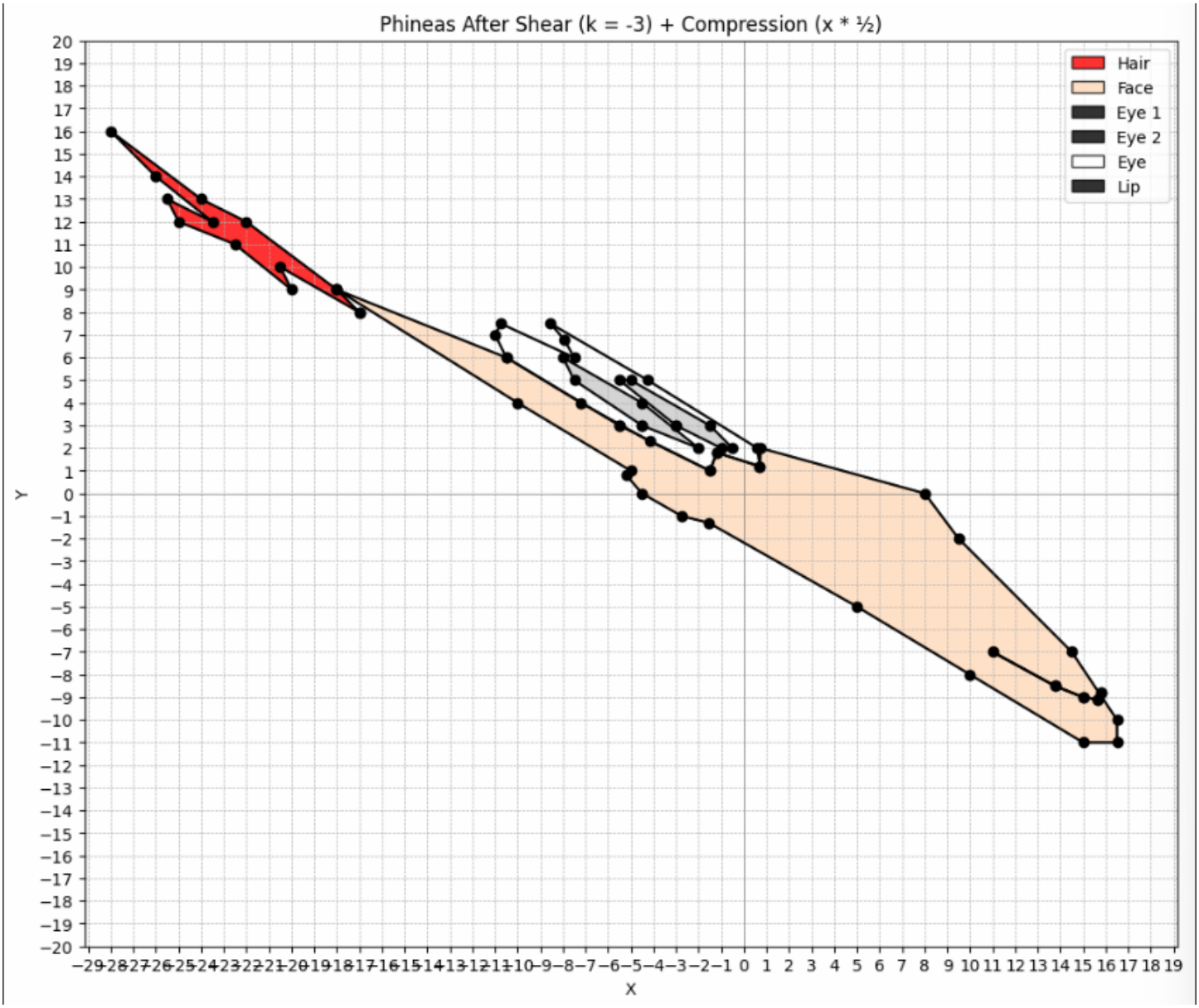
Sheared Figure (k = -3)



Compressed Figure



Final Sheared + Compressed Figure


Phineas After Shear (k = -3) + Compression (x * ½)
Hair
Face
Eye 1
Eye 2
Eye
Lip
X
Y