EXPERIMENT 2 - CREATE A 3D MODEL FROM 2D IMAGES

AIM:

To create a 3d model from 2d images

ALGORITHM:

1. Install Required Libraries

Ensure the necessary libraries matplotlib and pillow are installed.

2. Import Libraries

Import numpy, Image from PIL, and pyplot from matplotlib.
Import Axes3D from mpl_toolkits.mplot3d.

3. Load the Image

Define the path to the image file.
Load the image using Image.open.

4. Convert Image to Numpy Array

Convert the loaded image into a numpy array using np.array.

5. Create x and y Coordinates

Generate linearly spaced values for x and y coordinates using np.linspace.
Create a meshgrid for x and y coordinates using np.meshgrid.

6. Normalise the z Data

Compute the mean across the colour channels to get the intensity values for z.

7. Normalised RGB Values

Normalise the RGB values of the image array to the range [0, 1].

8. Create the 3D Plot

Initialise a 3D plot using fig.add_subplot.
Plot the surface using plot_surface with normalised image data as face colours.

9. Set Labels

Label the x, y, and z axes.

10. Show the Plot

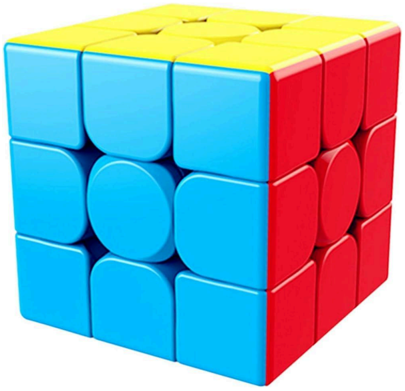
Display the plot using plt.show.

CODE:

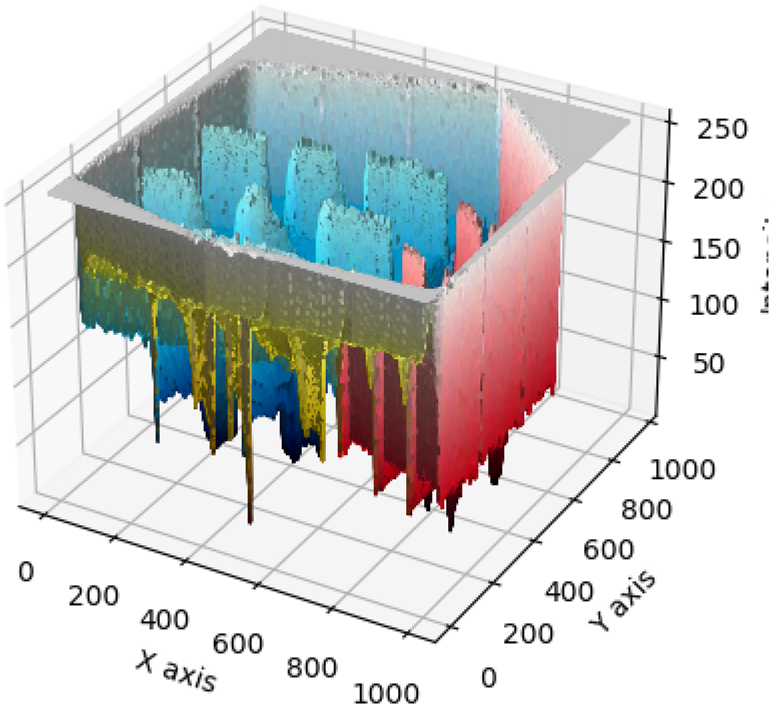
```
!pip install matplotlib pillow
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
```

```python
from mpl_toolkits.mplot3d import Axes3D

# Load the 2D image
image_path = '/content/ruby.jpg'  # Replace with your image file path
image = Image.open(image_path)  # Keep it in color

# Convert the 2D image to a numpy array
image_array = np.array(image)

# Create the x, y coordinates
x = np.linspace(0, image_array.shape[1], image_array.shape[1])
y = np.linspace(0, image_array.shape[0], image_array.shape[0])
x, y = np.meshgrid(x, y)

# Normalize the z data
z = np.mean(image_array, axis=2)

# Normalize RGB values to [0, 1]
norm_image_array = image_array / 255.0

# Create the 3D plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(x, y, z, facecolors=norm_image_array, rstride=1, cstride=1)

# Set labels
ax.set_xlabel('X axis')
ax.set_ylabel('Y axis')
ax.set_zlabel('Intensity')

# Show the plot
plt.show()
```

INPUT IMAGE:



OUTPUT IMAGE:



RESULT:

A 3d model from 2d images was created successfully.