EXPT 10	Motion Analysis (3D)

AIM:-

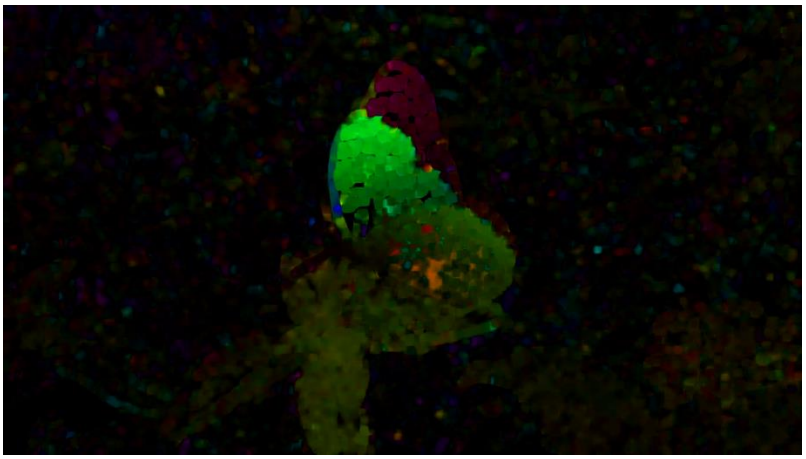
To implement dynamic stereo motion analysis for reconstructing 3D motion from stereo image pairs, processing both two-frame and multiple-frame sequences to estimate depth and track objects in three-dimensional space.

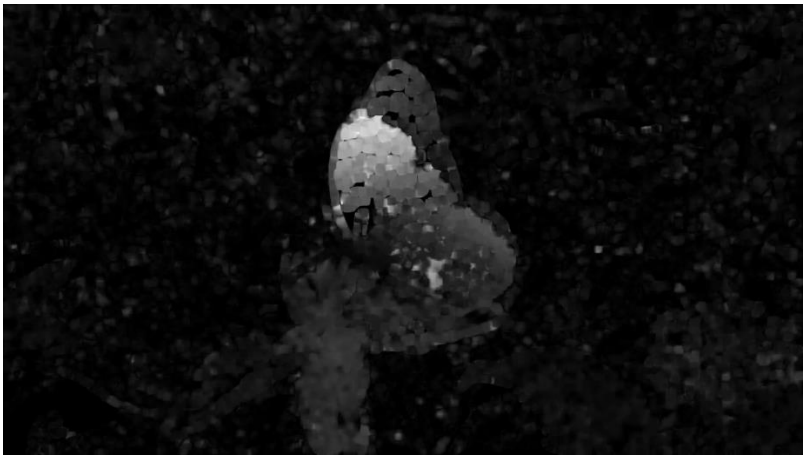
CODE:-

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow
from google.colab import files
import io

# Upload the video file
uploaded = files.upload()

# Get the filename
filename = list(uploaded.keys())[0]

cap = cv2.VideoCapture(filename)

ret, prev = cap.read()
prev_gray = cv2.cvtColor(prev, cv2.COLOR_BGR2GRAY)

mask = np.zeros_like(prev)

# Create a figure and subplots for displaying images
fig, axes = plt.subplots(2, 5, figsize=(20, 8))
axes = axes.flatten() # Flatten the 2x5 grid to a 1D array for easier indexing
```

```
count = 0
while True:
    ret, frame = cap.read()
    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    flow = cv2.calcOpticalFlowFarneback(prev_gray, gray, None,
                                        0.5, 3, 15, 3, 5, 1.2, 0)
    magnitude, angle = cv2.cartToPolar(flow[...,0], flow[...,1])
    mask[...,0] = angle * 180 / np.pi / 2
    mask[...,2] = cv2.normalize(magnitude, None, 0, 255, cv2.NORM_MINMAX)
    rgb = cv2.cvtColor(mask, cv2.COLOR_HSV2BGR)

    # Display the output in subplots
    if count < len(axes):
        axes[count].imshow(cv2.cvtColor(rgb, cv2.COLOR_BGR2RGB))
        axes[count].set_title(f'Frame {count}')
        axes[count].axis('off')
    count += 1

    prev_gray = gray

cap.release()
plt.tight_layout()
plt.show()
```

OUTPUT:-





RESULT:-

Successfully performed 3D motion analysis using stereo vision techniques, reconstructing depth information and tracking objects across multiple frames. The methods demonstrated the capability to analyze motion in three-dimensional space from stereo camera setups.