Lab Assignment 4

Problem Statement

Write a Python program to extract interest points of an image using the SIFT detector. Your program should load an image, detect keypoints using SIFT, and display the detected keypoints on the original image.

Hints and Approach

Libraries to Use

- **OpenCV:** For image processing, grayscale conversion, and applying the SIFT detector.
- **NumPy:** (Optional) For handling arrays if you need to manipulate image data further.
- **Matplotlib:** For visualizing the original and processed images.

Steps to Solve

1. **Load the Image:** Use OpenCV's `cv2.imread()` function to read the image from the given path.
2. **Display the Original Image:** Use Matplotlib's `plt.imshow()` function to display the original image.
3. **Convert the Image to Grayscale:** Use OpenCV's `cv2.cvtColor()` function to convert the image to grayscale for feature detection.
4. **Initialize the SIFT Detector:** Use OpenCV's `cv2.SIFT_create()` to initialize the Scale-Invariant Feature Transform (SIFT) detector.
5. **Detect Keypoints:** Apply the SIFT detector's `detect()` method to find interest points in the grayscale image.
6. **Draw the Keypoints:** Use OpenCV's `cv2.drawKeypoints()` function to overlay the detected keypoints on the original image.
7. **Display the Image with Keypoints:** Use Matplotlib's `plt.imshow()` to visualize the final image with keypoints.

Hints for Functions to Use

- **For reading the image:** `cv2.imread()`
- **For displaying images:** `plt.imshow()`, `plt.axis('off')` to remove axis labels.
- **For grayscale conversion:** `cv2.cvtColor()` with the flag `cv2.COLOR_BGR2GRAY`.
- **For initializing SIFT:** `cv2.SIFT_create()`
- **For detecting keypoints:** `sift.detect()`
- **For drawing keypoints:** `cv2.drawKeypoints()`

In [1]:
```
# Download assignment files
!wget https://github.com/buntyke/vnr_dlcv2024_labs/releases/download/DLCVLab4/table
```

```
--2024-12-08 07:04:28--  https://github.com/buntyke/vnr_dlcv2024_labs/releases/dow
nload/DLCVLab4/table-image.jpeg
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e
65be/878811324/a519bc8c-305e-4e5b-982f-7d63829ca4f8?X-Amz-Algorithm=AWS4-HMAC-SHA2
56&X-Amz-Credential=releaseassetproduction%2F20241208%2Fus-east-1%2Fs3%2Faws4_requ
est&X-Amz-Date=20241208T070428Z&X-Amz-Expires=300&X-Amz-Signature=ac593a2dac73bed2
f73c2a73163eaddc81e0a4bfcaa789ca6beedab8ce5d272c&X-Amz-SignedHeaders=host&response
-content-disposition=attachment%3B%20filename%3Dtable-image.jpeg&response-content-
type=application%2Foctet-stream [following]
--2024-12-08 07:04:28--  https://objects.githubusercontent.com/github-production-r
elease-asset-2e65be/878811324/a519bc8c-305e-4e5b-982f-7d63829ca4f8?X-Amz-Algorithm
=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241208%2Fus-east-1%2
Fs3%2Faws4_request&X-Amz-Date=20241208T070428Z&X-Amz-Expires=300&X-Amz-Signature=a
c593a2dac73bed2f73c2a73163eaddc81e0a4bfcaa789ca6beedab8ce5d272c&X-Amz-SignedHeader
s=host&response-content-disposition=attachment%3B%20filename%3Dtable-image.jpeg&re
sponse-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.19
9.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.19
9.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 130976 (128K) [application/octet-stream]
Saving to: ‘table-image.jpeg’

table-image.jpeg    100%[===================>] 127.91K  --.-KB/s    in 0.03s

2024-12-08 07:04:28 (4.65 MB/s) - ‘table-image.jpeg’ saved [130976/130976]
```

In [2]:

```python
### WRITE CODE HERE ###
import cv2
from matplotlib import pyplot as plt

# Load the image
image_path = './table-image.jpeg'
img = cv2.imread(image_path)

# Display the image using Matplotlib
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.axis('off') # Turn off axis labels
plt.show()

# Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Initialize the SIFT detector
sift = cv2.SIFT_create()

# Detect SIFT keypoints
keypoints = sift.detect(gray, None)

# Draw the keypoints on the image
output_image = cv2.drawKeypoints(
    img, keypoints, None,
    (0, 0, 255), flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

# Display the image using Matplotlib
plt.imshow(cv2.cvtColor(output_image, cv2.COLOR_BGR2RGB))
plt.axis('off') # Turn off axis labels
plt.show()
```

In [ ]: