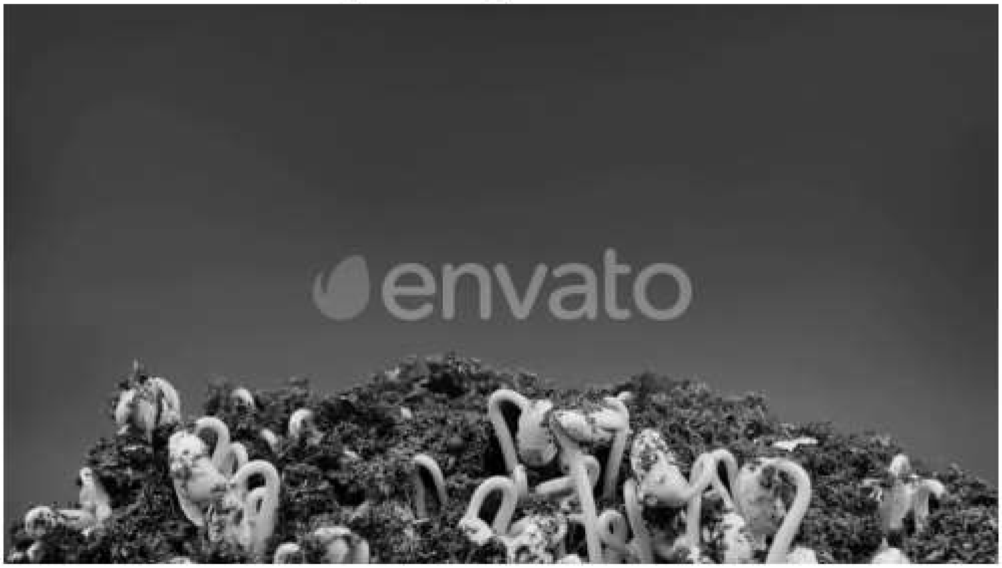
In [2]:

```python
import cv2
import matplotlib.pyplot as plt
import numpy as np

# Load the video
video_path = '/content/hmm.mp4'
cap = cv2.VideoCapture(video_path)

# Check if video opened successfully
if not cap.isOpened():
    print("Error opening video file")

# Create a folder to store extracted frames
import os
os.makedirs("frames", exist_ok=True)

frame_count = 0
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # Save the frame as an image
    frame_filename = f"frames/frame_{frame_count:04d}.jpg"
    cv2.imwrite(frame_filename, frame)
    frame_count += 1

cap.release()
```

In [3]:

```python
# Read an example frame for processing
frame = cv2.imread("frames/frame_0020.jpg")
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# Display the original grayscale frame
plt.imshow(gray, cmap='gray')
plt.title("Original Grayscale Frame")
plt.axis("off")
plt.show()
```

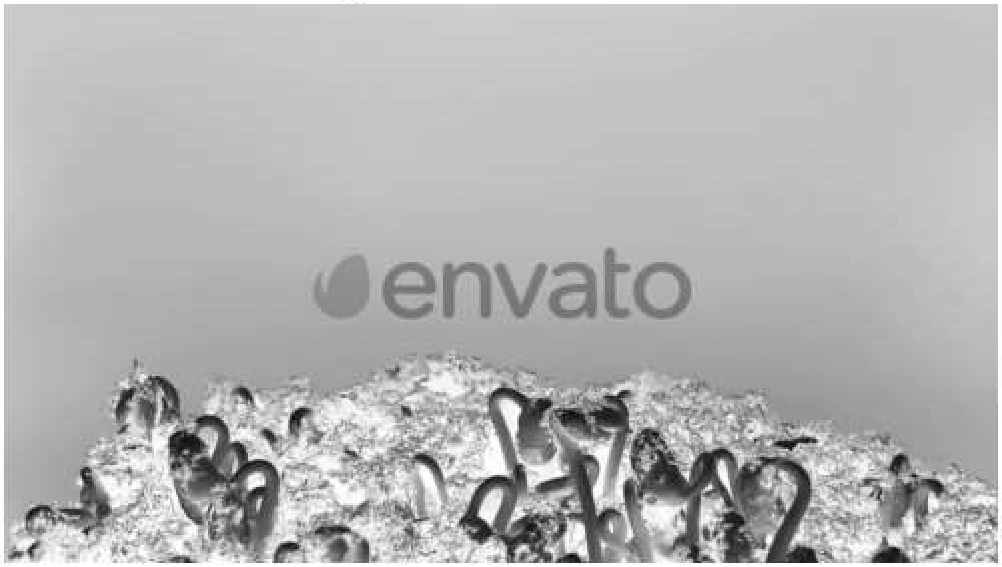
Original Grayscale Frame



In [4]:

```python
# Negative transformation
negative = 255 - gray

plt.imshow(negative, cmap='gray')
plt.title("Negative Transformation")
plt.axis("off")
plt.show()
```

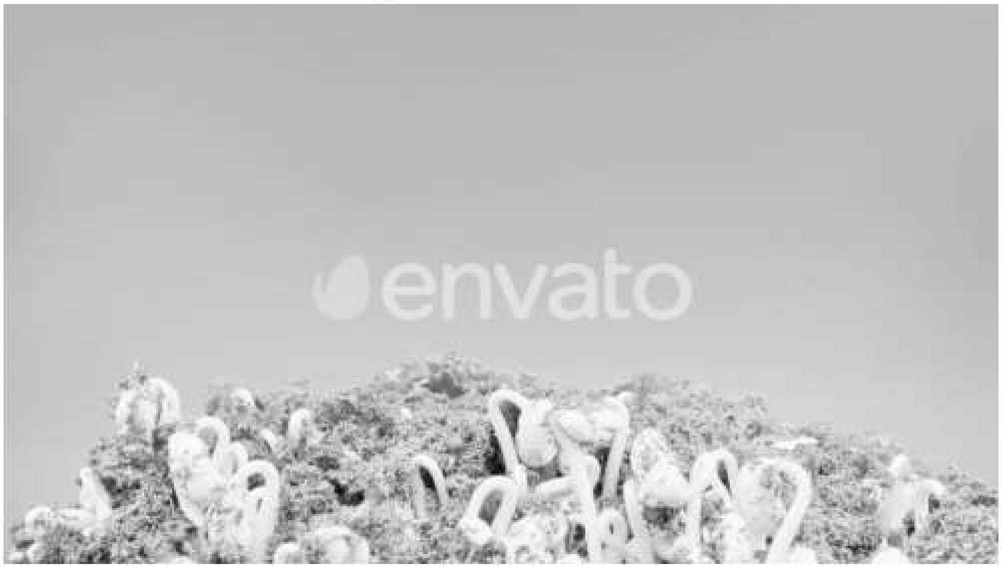
Negative Transformation



In [5]:

```python
# Log transformation
c = 255 / np.log(1 + np.max(gray))
log_transformed = c * np.log(1 + gray.astype(np.float32))
log_transformed = np.array(log_transformed, dtype=np.uint8)

plt.imshow(log_transformed, cmap='gray')
plt.title("Log Transformation")
plt.axis("off")
plt.show()
```

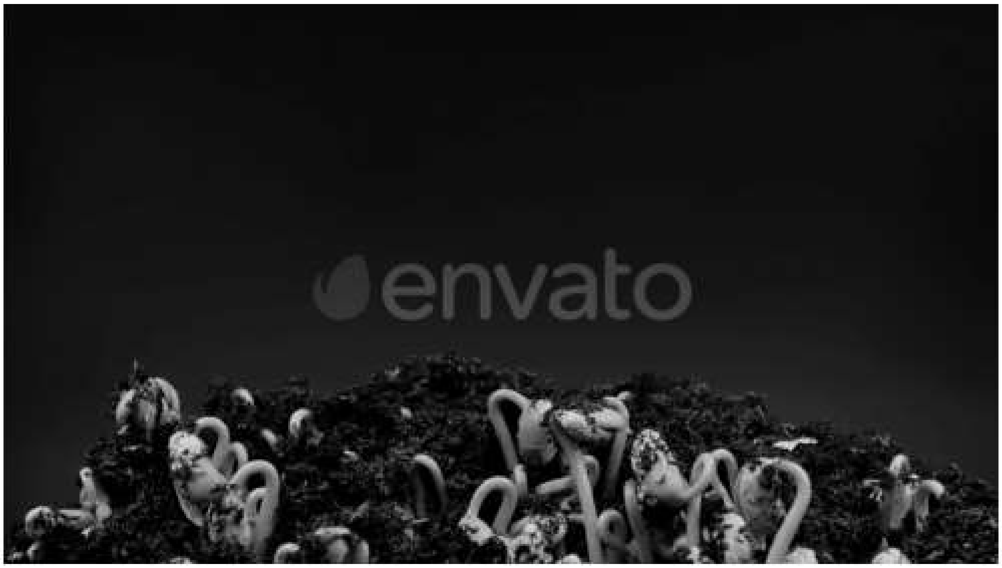
Log Transformation



In [6]:

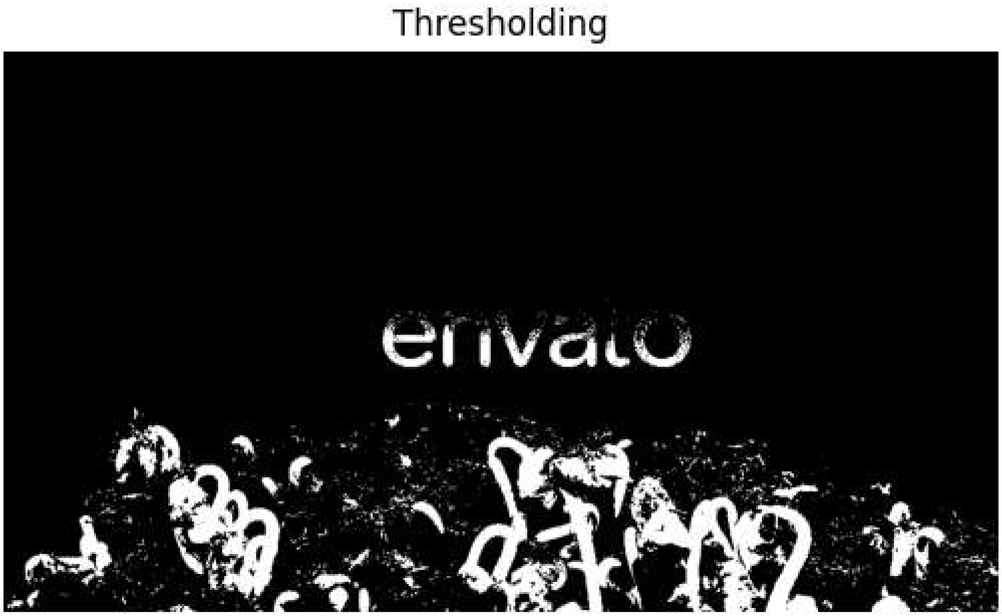
```python
# Gamma transformation
gamma = 2.0
gamma_corrected = np.array(255 * (gray / 255) ** gamma, dtype=np.uint8)

plt.imshow(gamma_corrected, cmap='gray')
plt.title("Gamma Transformation")
plt.axis("off")
plt.show()
```

Gamma Transformation



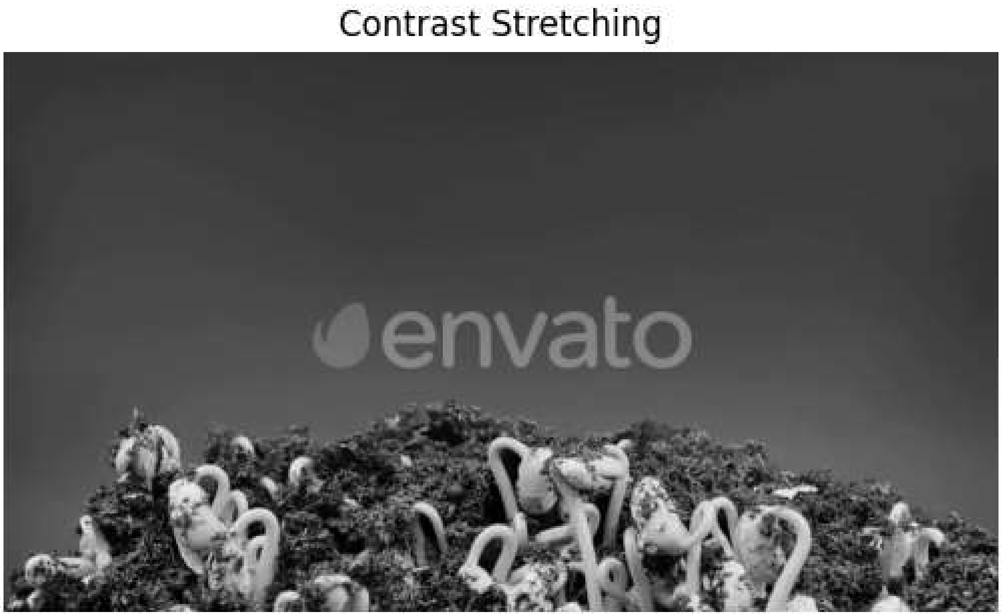
In [7]:

```python
# Thresholding
_, thresh_binary = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)

plt.imshow(thresh_binary, cmap='gray')
plt.title("Thresholding")
plt.axis("off")
plt.show()
```



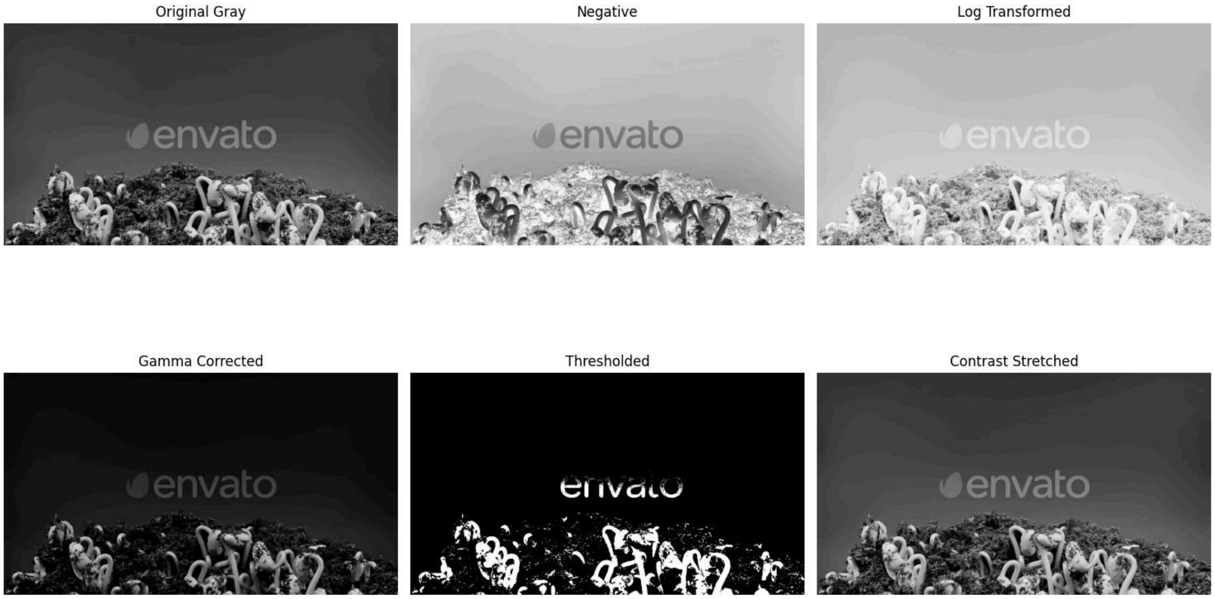
In [8]:

```
# Contrast stretching
min_val = np.min(gray)
max_val = np.max(gray)
contrast_stretched = ((gray - min_val) / (max_val - min_val) * 255).astype(np.uint8

plt.imshow(contrast_stretched, cmap='gray')
plt.title("Contrast Stretching")
plt.axis("off")
plt.show()
```



In [9]:

```
# Display all transformations together
titles = ['Original Gray', 'Negative', 'Log Transformed', 'Gamma Corrected',
          'Thresholded', 'Contrast Stretched']
images = [gray, negative, log_transformed, gamma_corrected,
          thresh_binary, contrast_stretched]

plt.figure(figsize=(15, 10))
for i in range(6):
    plt.subplot(2, 3, i + 1)
    plt.imshow(images[i], cmap='gray')
    plt.title(titles[i])
    plt.axis('off')

plt.tight_layout()
plt.show()
```



Original Gray

Negative

Log Transformed

Gamma Corrected

Thresholded

Contrast Stretched