

```
In [5]:
```

```
import os
import cv2
import numpy as np
from skimage.metrics import structural_similarity as ssim
import matplotlib.pyplot as plt
from scipy.stats import pearsonr, spearmanr
# Load and preprocess images
def load_and_preprocess_images(ref_path, test_path):
    ref_image = cv2.imread(ref_path)
    test_image = cv2.imread(test_path)
    if ref_image is None or test_image is None:
        raise ValueError("Error: Unable to load images from {} or {}".format(ref_path, test_path))

    # Resize test image to match reference image dimensions
    test_image_resized = cv2.resize(test_image, (ref_image.shape[1], ref_image.shape[0]))

    # Convert to grayscale
    ref_gray = cv2.cvtColor(ref_image, cv2.COLOR_BGR2GRAY)
    test_gray = cv2.cvtColor(test_image_resized, cv2.COLOR_BGR2GRAY)

    return ref_image, test_image_resized, ref_gray, test_gray
```

```
# Compare Images using SSIM and pixel-wise differences
def compare_images(ref_gray, test_gray):
    # Calculate SSIM score
    similarity_index, _ = ssim(ref_gray, test_gray, full=True)
    print(f"Structural Similarity Index (SSIM): {similarity_index:.4f}")

    # Calculate pixel-wise differences
    diff_image = cv2.absdiff(ref_gray, test_gray)
    diff_image_normalized = cv2.normalize(diff_image, None, 0, 255, cv2.NORM_MINMAX).astype(np.uint8)

    return ref_image, test_image_resized, similarity_index
```

```
# Calculate SSIM threshold using statistical method (mean + std deviation)
def determine_ssim_threshold(ssim_scores):
    mean_ssim = np.mean(ssim_scores)
    std_ssim = np.std(ssim_scores)

    # Using mean + standard deviation as threshold
    threshold = mean_ssim + std_ssim
    print(f"SSIM Threshold for misfit-1.jpg: ({threshold:.4f})")

    return threshold
```

```
# Visualize the SSIM score distribution
def visualize_ssim_distribution(partial_ssim_scores, full_ssim_score, threshold=None):
    plt.figure(figsize=(12, 9))
    plt.hist(partial_ssim_scores, bins=20, alpha=0.7, color='red')
    plt.axvline(full_ssim_score, color='green', linestyle='dashed', linewidth=2, label="Mean SSIM (Partial Insertion Magnets)")
    plt.legend()
    plt.xlabel('SSIM Score')
    plt.ylabel('Frequency')
    plt.show()
```

```
# 3D Histogram
fig = plt.figure(figsize=(12, 9))
ax = fig.add_subplot(111, projection='3d')
hist, bins = np.histogram(partial_ssim_scores, bins=20, density=True)
bin_centers = (bins[1:] - bins[1:]) / 2

ax.bar(bin_centers, hist, width=bins[1] - bins[0], alpha=0.7, color='red', label="Partial Insertion SSIMs")
mu, std = norm.fit(partial_ssim_scores)
x = np.linspace(mu - 3*std, mu + 3*std, 100)
p = norm.pdf(x, mu, std)
ax.plot(x, p, 'k', linewidth=2, label="Threshold ((threshold:.4f))")

ax.set_xlabel("SSIM Score")
ax.set_ylabel("Frequency")
ax.set_zlabel("Density")
ax.legend()
plt.show()
```

```
# Surface roughness analysis
def surface_roughness_analysis():
    # Load depth map
    depth_map = np.load(depth_map)
    surface_roughness = np.std(depth_map)
    print(f"Surface Roughness (Std Dev): {surface_roughness} (surface_roughness: .4f)")

    return surface_roughness
```

```
# Feature matching
def feature_matching(ref_gray, test_gray):
    orb = cv2.ORB_create()
    keypoints_ref, descriptors_ref = orb.detectAndCompute(ref_gray, None)
    keypoints_test, descriptors_test = orb.detectAndCompute(test_gray, None)
    bfMatcher = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
    matches = bfMatcher.match(descriptors_ref, descriptors_test)
    mask = np.zeros_like(ref_gray)
    drawMatches(mask, keypoints_ref, keypoints_test, matches)
    return matches, keypoints_ref, keypoints_test
```

```
# Depth analysis
def depth_analysis(ref_gray, test_gray):
    disparity_map = cv2.absdiff(ref_gray, test_gray)
    return disparity_map
```

```
# Calculate correlation metrics (Linear and Non-Linear)
def calculate_correlation(ref_gray, test_gray):
    ref_flat = ref_gray.flatten()
    test_flat = test_gray.flatten()

    Pearson_Correlation = pearsonr(ref_flat, test_flat)
    Spearman_NonLinear_Correlation = spearmanr(ref_flat, test_flat)
```

```
print(f"Linear Correlation (Pearson): {linear_corr:.4f}")
print(f"Non-linear Correlation (Spearman): {nonlinear_corr:.4f}")

return linear_corr, nonlinear_corr
```

```
# Magnet insertion percentage
def determine_magnet_insertion(test_gray):
    threshold = 50 # Threshold to identify inserted magnets
    inserted_pixels = np.sum(test_gray < threshold)
    total_pixels = test_gray.size
    insertion_percentage = (inserted_pixels / total_pixels) * 100
    print(f"Assembly Insertion Percentage: {insertion_percentage:.2f}%")
    return insertion_percentage
```

```
# Calculate alignment score
def calculate_alignment(score):
    alignment_score = score * 100
    print(f"Alignment Score: ({alignment_score:.2f}%)")
    return alignment_score
```

```
# Decision-making system
def decision_making(ssim_index, insertion_percentage, surface_roughness):
    threshold_ssim = 0.9
    threshold_insertion = 95.0
    threshold_roughness = 95.0

    if ssim_index < threshold_ssim and insertion_percentage < threshold_insertion:
        print(f"Decision: Assembly issue detected (low SSIM and magnet not fully inserted).")
    elif ssim_index > threshold_ssim:
        print(f"Decision: Assembly issue detected (low SSIM).")
    elif insertion_percentage > threshold_insertion:
        print(f"Decision: Magnet not fully inserted.")
    elif surface_roughness > threshold_roughness:
        print(f"Decision: Surface quality issue detected (high surface roughness).")
    else:
        print(f"Decision: Assembly is acceptable.")
```

```
# Main execution function
def main():
    ref_path = "magnet_insertion_proper.jpg" # Path to the fully inserted magnet image
    input_folder = "new_images" # Folder containing test images (partially inserted magnets)
```

```
# Lists to store SSIM scores
full_insertion_ssims = []
partial_insertion_ssims = []

# Get list of images in the input folder
test_images = [f for f in os.listdir(input_folder) if f.endswith('.jpg', '.png')]
```

```
# Process each image
for test_image_name in test_images:
    print(f"Processing: {test_image_name}")
    print(f"Input path: {input_folder}/{test_image_name}")
    print(f"Output path: {output_folder}/{test_image_name}")

    # Load and preprocess images
    ref_image, test_image, ref_gray, test_gray = load_and_preprocess_images(ref_path, test_path)
```

```
# Compare Images
diff_image, ssim_index = compare_images(ref_gray, test_gray)
```

```
# Collect SSIM score for further analysis
if ssim_index < threshold_ssim:
    full_insertion_ssims.append(ssim_index)
else:
    partial_insertion_ssims.append(ssim_index)
```

```
# Calculate and display the threshold for the current image
current_threshold = determine_ssim_threshold(partial_insertion_ssims)
print(f"Current SSIM Threshold for {test_image_name}: ({current_threshold:.4f})")
```

```
# Feature matching
matches, keypoints_ref, keypoints_test = feature_matching(ref_gray, test_gray)
```

```
# Depth analysis
disparity_map = depth_analysis(ref_gray, test_gray)
```

```
# Surface roughness analysis
surface_roughness = surface_roughness_analysis()
```

```
# Linear and Non-linear Correlation
linear_corr, nonlinear_corr = calculate_correlation(ref_gray, test_gray)
```

```
print(f"Linear Correlation (Pearson): {linear_corr:.4f}")
print(f"Non-linear Correlation (Spearman): {nonlinear_corr:.4f}")

return linear_corr, nonlinear_corr
```

```
# Magnet insertion percentage
insertion_percentage = determine_magnet_insertion(test_gray)
```

```
# Alignment Score
alignment_score = calculate_alignment(score)
```

```
# Decision-making
decision_making(ssim_index, insertion_percentage, surface_roughness)
```

```
# Visualize results with larger images
plt.subplot(2, 3, 1) # Larger figure for better visibility
plt.title("Reference Image")
plt.imshow(cv2.cvtColor(ref_image, cv2.COLOR_BGR2RGB))
plt.subplot(2, 3, 2)
plt.title("Test Image")
plt.imshow(cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB))
plt.subplot(2, 3, 3)
plt.title("Pixel-wise Differences")
plt.imshow(disparity_map, cmap='jet')
plt.tight_layout()
plt.show()
```

```
# Determine the threshold for SSIM classification
ssim_threshold = determine_ssim_threshold(partial_insertion_ssims)
```

```
# Compute SSIM score for fully inserted magnet (reference image)
fully_inserted_image = load_and_preprocess_images(ref_path, ref_path)[1]
```

```
full_ssim_score = compare_images(fully_inserted_image, fully_inserted_image)[1]
```

```
# Visualize SSIM score distribution
visualize_ssim_distribution(partial_insertion_ssims, full_ssim_score, threshold=ssim_threshold)
```

```
print(f"SSIM score for fully inserted magnet: {full_ssim_score:.4f}")
```

```
print(f"SSIM score for the fully inserted magnet falls 'above' if full_ssim_score > ssim_threshold else 'below' the threshold.")
```

```
if __name__ == "__main__":
    main()
```

```
Processing: misfit-1.jpg
Structural Similarity Index (SSIM): 0.8701
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for misfit-1.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.6312
Non-linear Correlation (Spearman): 0.5780
Magnet Insertion Percentage: 1.45%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Processing: misfit-2.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for misfit-2.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.6312
Non-linear Correlation (Spearman): 0.5780
Magnet Insertion Percentage: 1.45%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Processing: misfit-2.1.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for misfit-2.1.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.6312
Non-linear Correlation (Spearman): 0.5780
Magnet Insertion Percentage: 1.45%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Processing: misfit-2.2.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for misfit-2.2.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.6312
Non-linear Correlation (Spearman): 0.5780
Magnet Insertion Percentage: 1.45%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Processing: misfit-2.3.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for misfit-2.3.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.6312
Non-linear Correlation (Spearman): 0.5780
Magnet Insertion Percentage: 1.45%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Processing: misfit-3.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for misfit-3.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.6312
Non-linear Correlation (Spearman): 0.5780
Magnet Insertion Percentage: 1.45%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Processing: misfit-3.1.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for misfit-3.1.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.6312
Non-linear Correlation (Spearman): 0.5780
Magnet Insertion Percentage: 1.45%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Processing: misfit-3.2.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for misfit-3.2.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.6312
Non-linear Correlation (Spearman): 0.5780
Magnet Insertion Percentage: 1.45%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Processing: misfit-3.3.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for misfit-3.3.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.6312
Non-linear Correlation (Spearman): 0.5780
Magnet Insertion Percentage: 1.45%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Processing: misfit-3.4.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for misfit-3.4.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.6312
Non-linear Correlation (Spearman): 0.5780
Magnet Insertion Percentage: 1.45%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```