

Advanced Spectral Imaging: Homework 4

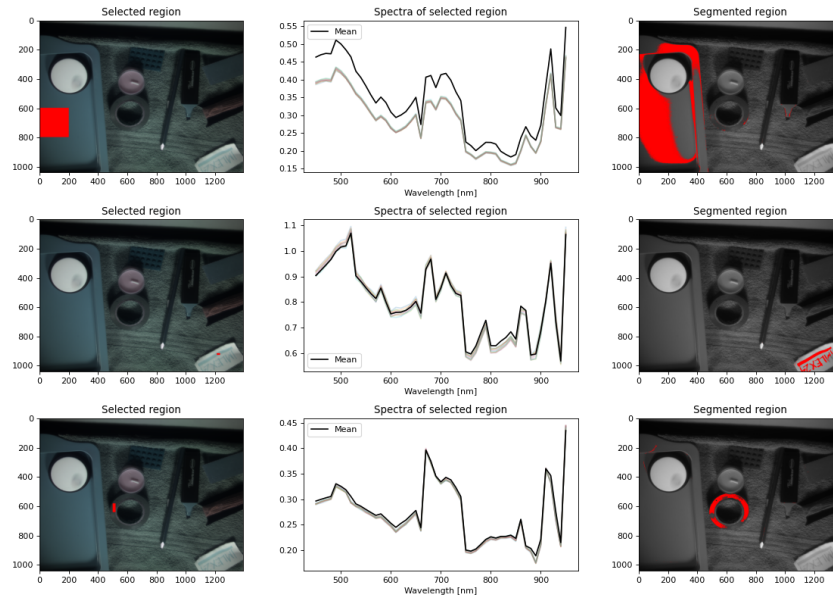
Yuya Haga

December 28, 2024

1 Task #1. Segmentation of green materials

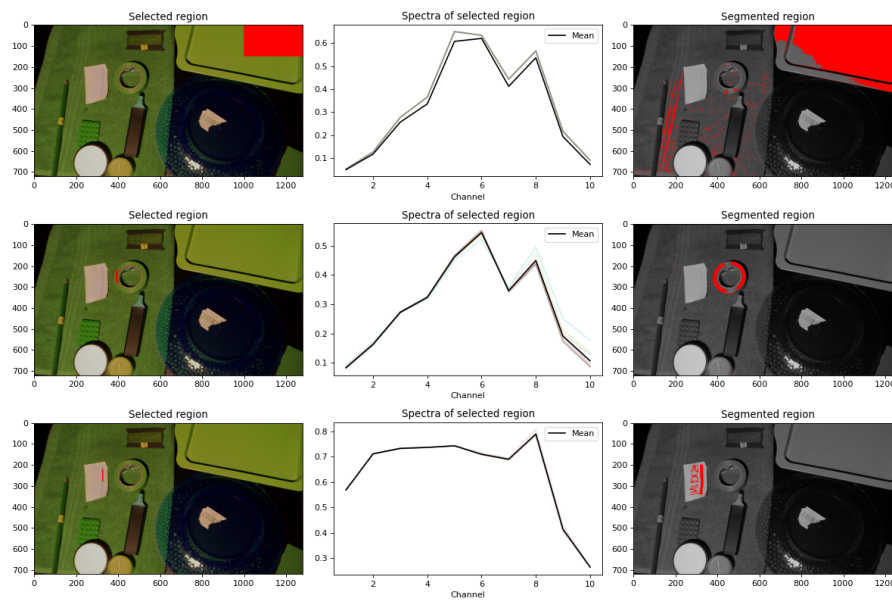
Segmentation results of green materials by Nuance camera is shown in Figure 1. Codes for this task can be found in Code 1.

Figure 1: Segmentation results by Nuance camera



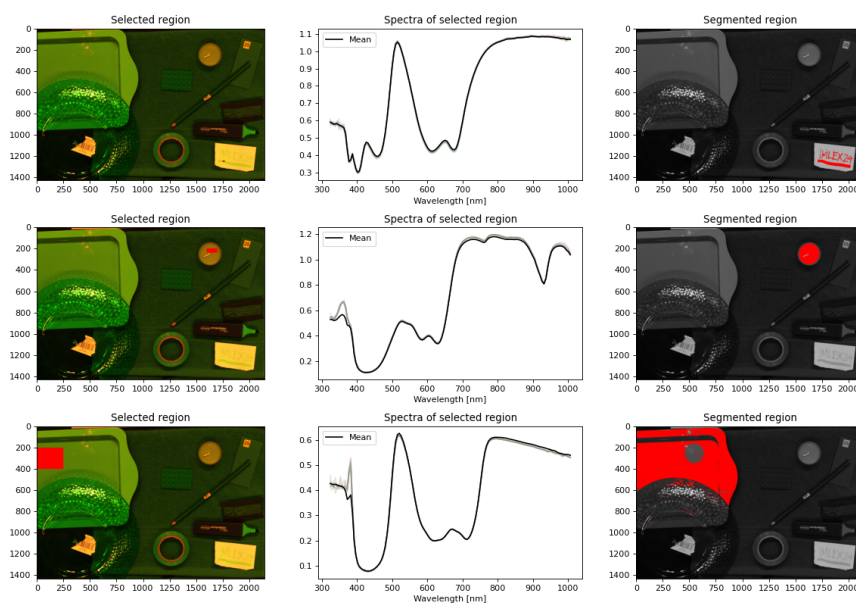
Segmentation results of green materials by Tuneble light camera is shown in Figure 2. Codes for this task can be found in Code 2.

Figure 2: Segmentation results by Tunable light camera



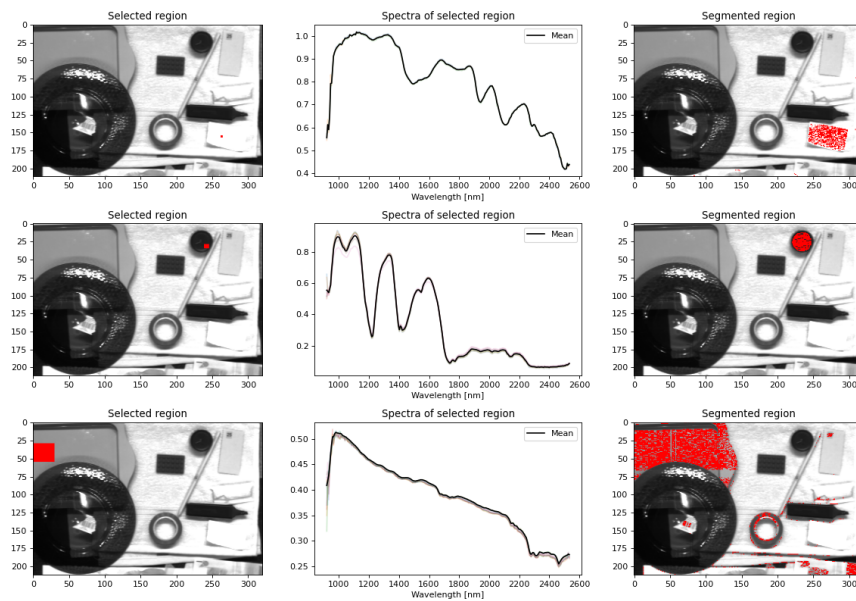
Segmentation results of green materials by Specim Scanner in visible wavelengths is shown in Figure 3. Codes for this task can be found in Code 3.

Figure 3: Segmentation results by Specim Scanner in visible wavelengths



Segmentation results of green materials by Specim Scanner in infrared wavelengths is shown in Figure 4. Codes for this task can be found in Code 4.

Figure 4: Segmentation results by Specim Scanner in infrared wavelengths



2 Task #2. Plastic green leaves

c

3 Task #3. powders / medical image

c

4 Codes

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 from asi import path_config
5 from asi.draw import reconstruct_rgb
6 from asi.io.load_nuance import load_nuance_image
7 from segmentation import plot_segmentation_results
8
9 # Load spectral image and apply white correction
10 session2 = path_config.measurements / "session2"
11 nuance = session2 / "Nuance"
12 root = nuance / "greenmaterials"
13
14 # load spectral image
15 spectral_image, wavelengths = load_nuance_image(root)
16 spectral_image = spectral_image.astype(np.float64)
17
18 white_pos = (slice(320, 400), slice(100, 200))
19
20 # White correction with selected area
21 white_sq = spectral_image[white_pos]
22

```

```
23 # replace nonzero elements with minimum value
24 nonzero_elements = white_sq[white_sq != 0]
25 min_elm = nonzero_elements.min()
26 white_sq = white_sq.clip(min_elm, None)
27
28 # apply white correction
29 whiteref = white_sq.mean(axis=(0, 1))
30 spectral_image /= whiteref
31
32 rgb_view = reconstruct_rgb(spectral_image, wavelengths)
33
34
35 fig, axes = plt.subplots(3, 3, tight_layout=True, figsize=(15, 10), dpi=80)
36
37 select_pos = (slice(600, 800), slice(0, 200))
38 plot_segmentation_results(
39     axes[0],
40     spectral_image,
41     wavelengths,
42     select_pos,
43     rgb_view,
44     threshold=0.11,
45 )
46
47 select_pos = (slice(915, 927), slice(1212, 1235))
48 plot_segmentation_results(
49     axes[1],
50     spectral_image,
51     wavelengths,
52     select_pos,
53     rgb_view,
54     threshold=0.2,
55 )
56 select_pos = (slice(580, 640), slice(500, 520))
57 plot_segmentation_results(
58     axes[2],
59     spectral_image,
60     wavelengths,
61     select_pos,
62     rgb_view,
63     threshold=0.080,
64 )
65
66 fig.show()
67 fig.savefig("./fig/task1/nuance.png")
```

Code 1: Segmentation of green materials by Nuance camera

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 from asi import path_config
5 from asi.io.load_tunable import load_tunable_image
6 from segmentation import plot_segmentation_results
7
8 # Configurations
9 WHITE_POS = (slice(600, 700), slice(230, 300))
10 # LOAD IMAGE
```

```
11 session1 = path_config.measurements / "session2"
12 tunable_root = session1 / "Tunable" / "green materials"
13
14 spectral_image, channels = load_tunable_image(
15     tunable_root,
16     name="colorchecker",
17     white_pos=WHITE_POS,
18 )
19 spectral_image = spectral_image.astype(np.float64)
20
21 # Apply white correction
22 white_sq = spectral_image[WHITE_POS]
23 whiteref = white_sq.mean(axis=(0, 1))
24 white_corrected = spectral_image / whiteref
25 spectral_image /= spectral_image.max()
26
27
28 # Make RGB view
29 SELECT_CHANNELS = [7, 5, 1]
30 # Make RGB view
31 rgb_view = spectral_image[..., SELECT_CHANNELS]
32 # Postprocess for preview
33 rgb_view /= rgb_view.max()
34 rgb_view *= 0.8
35 rgb_view = rgb_view.clip(0, 1)
36
37 wavelengths = channels
38
39 fig, axes = plt.subplots(3, 3, tight_layout=True, figsize=(15, 10), dpi=80)
40
41
42 select_pos = (slice(0, 150), slice(1000, None))
43 plot_segmentation_results(
44     axes[0],
45     spectral_image,
46     wavelengths,
47     select_pos,
48     rgb_view,
49     threshold=0.1,
50 )
51
52 select_pos = (slice(220, 280), slice(390, 400))
53 plot_segmentation_results(
54     axes[1],
55     spectral_image,
56     wavelengths,
57     select_pos,
58     rgb_view,
59     threshold=0.05,
60 )
61
62 select_pos = (slice(220, 280), slice(325, 330))
63 plot_segmentation_results(
64     axes[2],
65     spectral_image,
66     wavelengths,
67     select_pos,
68     rgb_view,
```

```

69     threshold=0.11,
70 )
71
72 fig.show()
73 fig.savefig("./fig/task1/tunable.png")

```

Code 2: Segmentation of green materials by Tunable light camera

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 from asi import path_config
5 from asi.draw import reconstruct_rgb_envi
6 from asi.preprocess import load_white_corrected
7 from asi.utils import get_wavelengths
8 from segmentation import plot_segmentation_results
9
10 # Load spectral image and apply white correction
11 session2 = path_config.measurements / "session2"
12 specim_scanner = session2 / "Specim scanner" / "GreenSamplesVisible" / "capture"
13
14 name = "solutions_scan_0145"
15 image_path = specim_scanner / name
16 darkref_path = specim_scanner / f"DARKREF_{name}"
17 whiteref_path = specim_scanner / f"WHITEREF_{name}"
18
19 spectral_image, envi_header = load_white_corrected(
20     image_path,
21     whiteref_path,
22     darkref_path,
23 )
24 spectral_image = spectral_image.astype(np.float16)
25
26 rgb_view = reconstruct_rgb_envi(spectral_image, envi_header)
27 rgb_view *= 3
28 rgb_view = rgb_view.clip(0, 1)
29 wavelengths = get_wavelengths(envi_header)
30
31
32 fig, axes = plt.subplots(3, 3, tight_layout=True, figsize=(15, 10), dpi=80)
33
34
35 select_pos = (slice(1245, 1260), slice(1750, 1770))
36 plot_segmentation_results(
37     axes[0],
38     spectral_image,
39     wavelengths,
40     select_pos,
41     rgb_view,
42     threshold=0.3,
43 )
44 select_pos = (slice(200, 240), slice(1600, 1700))
45 plot_segmentation_results(
46     axes[1],
47     spectral_image,
48     wavelengths,
49     select_pos,
50     rgb_view,

```

```

51     threshold=0.3,
52 )
53 select_pos = (slice(200, 400), slice(0, 250))
54 plot_segmentation_results(
55     axes[2],
56     spectral_image,
57     wavelengths,
58     select_pos,
59     rgb_view,
60 )
61
62 fig.show()
63 fig.savefig("./fig/task1/specim-scanner-vis.png")

```

Code 3: Segmentation of green materials by Specim Scanner in visible wavelengths

```

1
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 from asi import path_config
6 from asi.draw import reconstruct_gray_view
7 from asi.preprocess import load_white_corrected
8 from asi.utils import get_wavelengths
9 from segmentation import plot_segmentation_results
10
11 # Load spectral image and apply white correction
12 session2 = path_config.measurements / "session2"
13 specim_scanner = session2 / "Specim scanner" / "GreenSamplesIR" / "capture"
14
15 name = "IR_scan_0463"
16 image_path = specim_scanner / name
17 darkref_path = specim_scanner / f"DARKREF_{name}"
18 whiteref_path = specim_scanner / f"WHITEREF_{name}"
19
20 spectral_image, envi_header = load_white_corrected(
21     image_path,
22     whiteref_path,
23     darkref_path,
24 )
25 spectral_image = spectral_image.astype(np.float16)
26
27 rgb_view = reconstruct_gray_view(spectral_image)
28 rgb_view *= 1.5
29 rgb_view = rgb_view.clip(0, 1)
30 wavelengths = get_wavelengths(envi_header)
31
32
33 fig, axes = plt.subplots(3, 3, tight_layout=True, figsize=(15, 10), dpi=80)
34
35
36 select_pos = (slice(155, 158), slice(261, 264))
37 plot_segmentation_results(
38     axes[0],
39     spectral_image,
40     wavelengths,
41     select_pos,
42     rgb_view,

```

```
43     threshold=0.13,  
44 )  
45 select_pos = (slice(29, 35), slice(238, 245))  
46 plot_segmentation_results(  
47     axes[1],  
48     spectral_image,  
49     wavelengths,  
50     select_pos,  
51     rgb_view,  
52     threshold=0.15,  
53 )  
54 select_pos = (slice(29, 55), slice(0, 30))  
55 plot_segmentation_results(  
56     axes[2],  
57     spectral_image,  
58     wavelengths,  
59     select_pos,  
60     rgb_view,  
61     threshold=0.15,  
62 )  
63  
64 fig.show()  
65 fig.savefig("../fig/task1/specium-scanner-ir.png")
```

Code 4: Segmentation of green materials by Specim Scanner in infrared wavelengths