# Advanced Spectral Imaging: Homework 4

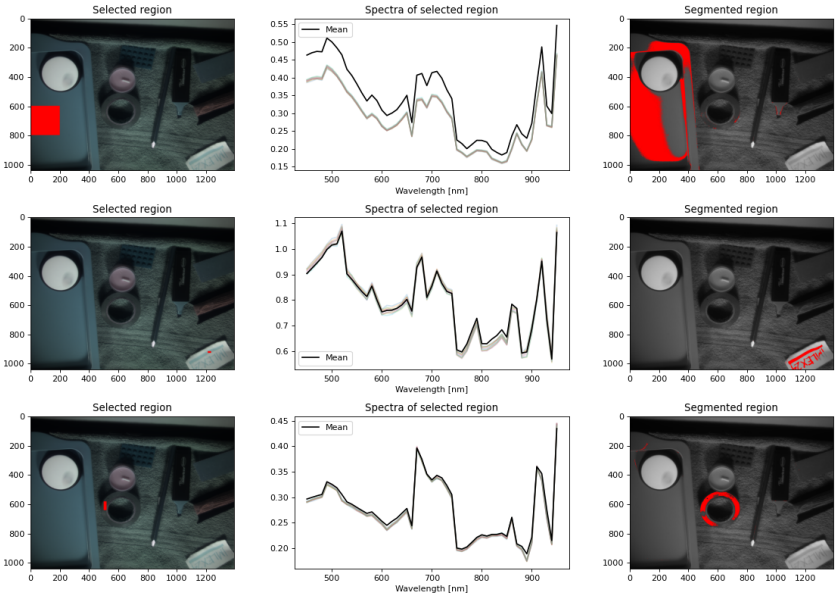
Yuya Haga

December 28, 2024

## 1 Task #1. Segmentation of green materials

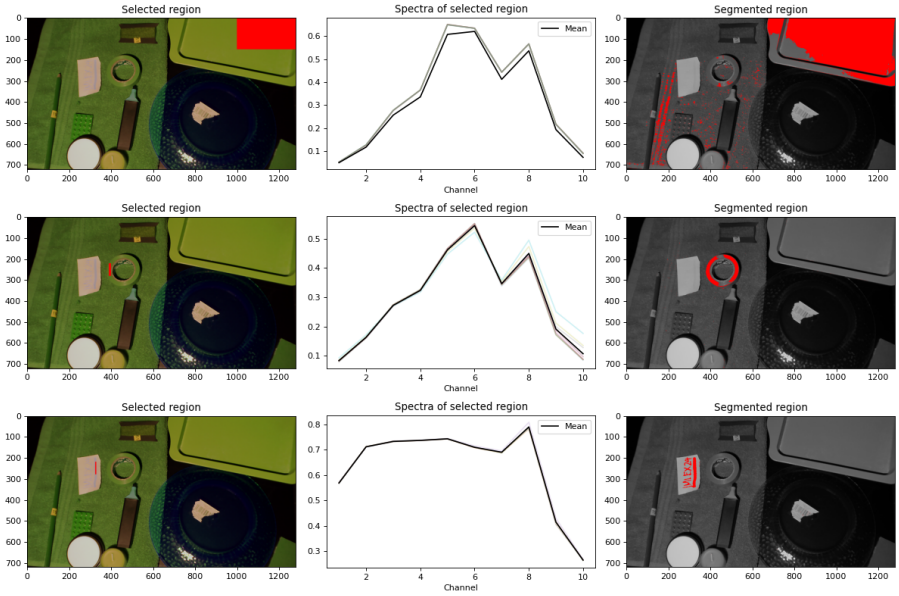
Segmentation results of green materials by Nuance camera is shown in Figure 1. Codes for this task can be found in Code 1.

Figure 1: Segmentation results by Nuance camera



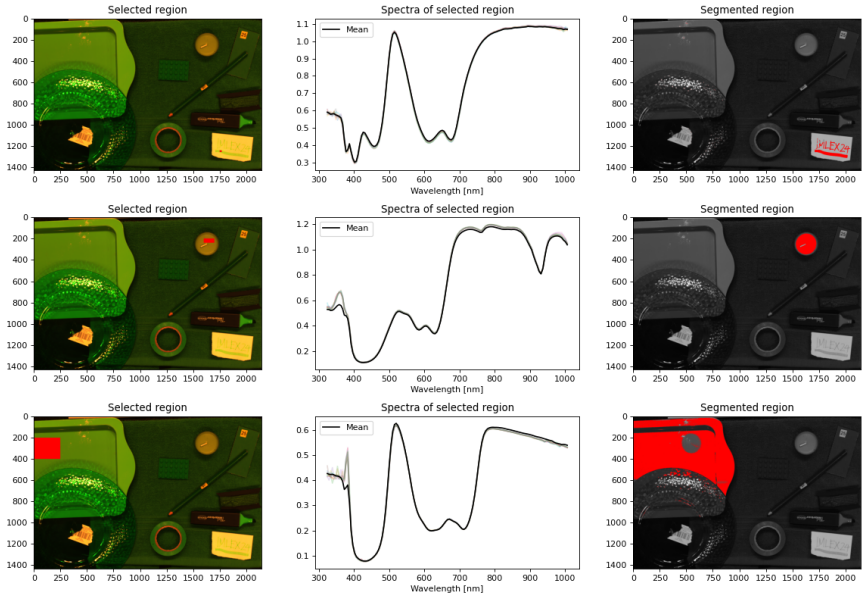

Segmentation results of green materials by Tuneble light camera is shown in Figure 2. Codes for this task can be found in Code 2.

Figure 2: Segmentation results by Tunable light camera



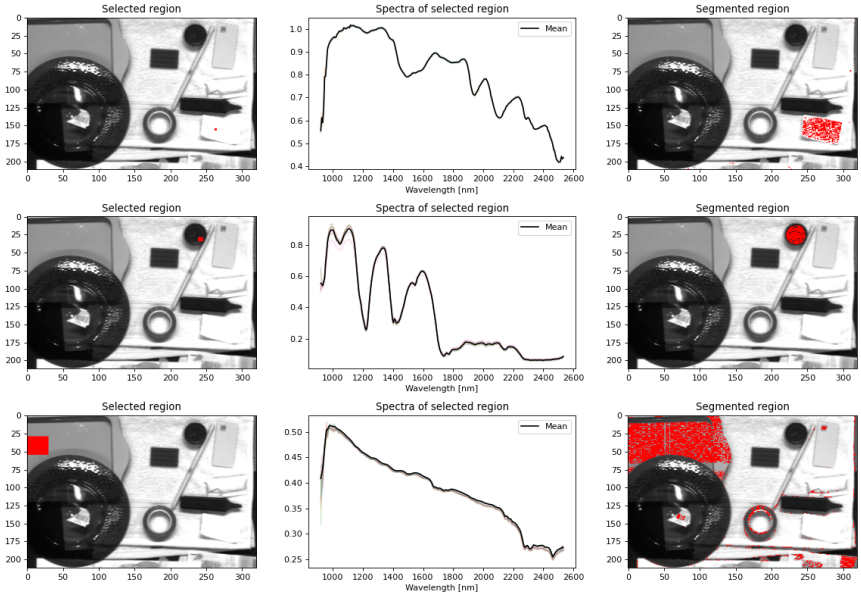

Segmentation results of green materials by Specim Scanner in visible wavelengths is shown in Figure 3. Codes for this task can be found in Code 3.

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

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

from asi import path_config
from asi.draw import reconstruct_rgb
from asi.io.load_nuance import load_nuance_image
from segmentation import plot_segmentation_results

# Load spectral image and apply white correction
session2 = path_config.measurements / "session2"
nuance = session2 / "Nuance"
root = nuance / "greenmaterials"

# load spectral image
spectral_image, wavelengths = load_nuance_image(root)
spectral_image = spectral_image.astype(np.float64)

white_pos = (slice(320, 400), slice(100, 200))

# White correction with selected area
white_sq = spectral_image[white_pos]

```

```
# replace nonzero elements with minimum value
nonzero_elements = white_sq[white_sq != 0]
min_elm = nonzero_elements.min()
white_sq = white_sq.clip(min_elm, None)

# apply white correction
whiteref = white_sq.mean(axis=(0, 1))
spectral_image /= whiteref

rgb_view = reconstruct_rgb(spectral_image, wavelengths)


fig, axes = plt.subplots(3, 3, tight_layout=True, figsize=(15, 10), dpi=80)

select_pos = (slice(600, 800), slice(0, 200))
plot_segmentation_results(
    axes[0],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
    threshold=0.11,
)

select_pos = (slice(915, 927), slice(1212, 1235))
plot_segmentation_results(
    axes[1],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
    threshold=0.2,
)
select_pos = (slice(580, 640), slice(500, 520))
plot_segmentation_results(
    axes[2],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
    threshold=0.080,
)

fig.show()
fig.savefig("./fig/task1/nuance.png")
```

Code 1: Segmentation of green materials by Nuance camera

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

from asi import path_config
from asi.io.load_tunable import load_tunable_image
from segmentation import plot_segmentation_results

# Configurations
WHITE_POS = (slice(600, 700), slice(230, 300))
# LOAD IMAGE
```

```
session1 = path_config.measurements / "session2"
tunable_root = session1 / "Tunable" / "green materials"

spectral_image, channels = load_tunable_image(
    tunable_root,
    name="colorchecker",
    white_pos=WHITE_POS,
)
spectral_image = spectral_image.astype(np.float64)

# Apply white correction
white_sq = spectral_image[WHITE_POS]
whiteref = white_sq.mean(axis=(0, 1))
white_corrected = spectral_image / whiteref
spectral_image /= spectral_image.max()


# Make RGB view
SELECT_CHANNELS = [7, 5, 1]
# Make RGB view
rgb_view = spectral_image[..., SELECT_CHANNELS]
# Postprocess for preview
rgb_view /= rgb_view.max()
rgb_view *= 0.8
rgb_view = rgb_view.clip(0, 1)

wavelengths = channels

fig, axes = plt.subplots(3, 3, tight_layout=True, figsize=(15, 10), dpi=80)


select_pos = (slice(0, 150), slice(1000, None))
plot_segmentation_results(
    axes[0],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
    threshold=0.1,
)

select_pos = (slice(220, 280), slice(390, 400))
plot_segmentation_results(
    axes[1],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
    threshold=0.05,
)

select_pos = (slice(220, 280), slice(325, 330))
plot_segmentation_results(
    axes[2],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
```

```
    threshold=0.11,
)

fig.show()
fig.savefig("./fig/task1/tunable.png")
```

Code 2: Segmentation of green materials by Tunable light camera

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

from asi import path_config
from asi.draw import reconstruct_rgb_envi
from asi.preprocess import load_white_corrected
from asi.utils import get_wavelengths
from segmentation import plot_segmentation_results

# Load spectral image and apply white correction
session2 = path_config.measurements / "session2"
specim_scanner = session2 / "Specim scanner" / "GreenSamplesVisible" / "capture"

name = "solutions_scan_0145"
image_path = specim_scanner / name
darkref_path = specim_scanner / f"DARKREF_{name}"
whiteref_path = specim_scanner / f"WHITEREF_{name}"

spectral_image, envi_header = load_white_corrected(
    image_path,
    whiteref_path,
    darkref_path,
)
spectral_image = spectral_image.astype(np.float16)

rgb_view = reconstruct_rgb_envi(spectral_image, envi_header)
rgb_view *= 3
rgb_view = rgb_view.clip(0, 1)
wavelengths = get_wavelengths(envi_header)


fig, axes = plt.subplots(3, 3, tight_layout=True, figsize=(15, 10), dpi=80)


select_pos = (slice(1245, 1260), slice(1750, 1770))
plot_segmentation_results(
    axes[0],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
    threshold=0.3,
)
select_pos = (slice(200, 240), slice(1600, 1700))
plot_segmentation_results(
    axes[1],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
```

```
    threshold=0.3,
)
select_pos = (slice(200, 400), slice(0, 250))
plot_segmentation_results(
    axes[2],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
)

fig.show()
fig.savefig("./fig/task1/specium-scanner-vis.png")
```

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

```

import matplotlib.pyplot as plt
import numpy as np

from asi import path_config
from asi.draw import reconstruct_gray_view
from asi.preprocess import load_white_corrected
from asi.utils import get_wavelengths
from segmentation import plot_segmentation_results

# Load spectral image and apply white correction
session2 = path_config.measurements / "session2"
specim_scanner = session2 / "Specim scanner" / "GreenSamplesIR" / "capture"

name = "IR_scan_0463"
image_path = specim_scanner / name
darkref_path = specim_scanner / f"DARKREF_{name}"
whiteref_path = specim_scanner / f"WHITEREF_{name}"

spectral_image, envi_header = load_white_corrected(
    image_path,
    whiteref_path,
    darkref_path,
)
spectral_image = spectral_image.astype(np.float16)

rgb_view = reconstruct_gray_view(spectral_image)
rgb_view *= 1.5
rgb_view = rgb_view.clip(0, 1)
wavelengths = get_wavelengths(envi_header)


fig, axes = plt.subplots(3, 3, tight_layout=True, figsize=(15, 10), dpi=80)


select_pos = (slice(155, 158), slice(261, 264))
plot_segmentation_results(
    axes[0],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
```

```
    threshold=0.13,
)
select_pos = (slice(29, 35), slice(238, 245))
plot_segmentation_results(
    axes[1],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
    threshold=0.15,
)
select_pos = (slice(29, 55), slice(0, 30))
plot_segmentation_results(
    axes[2],
    spectral_image,
    wavelengths,
    select_pos,
    rgb_view,
    threshold=0.15,
)

fig.show()
fig.savefig("./fig/task1/specium-scanner-ir.png")
```

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