In [ ]:
```
import pandas as _hex_pandas
import datetime as _hex_datetime
import json as _hex_json
```

In [ ]:
```
hex_scheduled = _hex_json.loads("false")
```

In [ ]:
```
hex_user_email = _hex_json.loads("\"example-user@example.com\"")
```

In [ ]:
```
hex_run_context = _hex_json.loads("\"logic\"")
```

In [ ]:
```
hex_timezone = _hex_json.loads("\"UTC\"")
```

In [ ]:
```
hex_project_id = _hex_json.loads("\"d1899f23-8c9d-4cff-9db3-60dfcf462629\"")
```

In [ ]:
```
hex_color_palette = _hex_json.loads("[\"#4C78A8\",\"#F58518\",\"#E45756\",\"#72B7B2\",\"#54A24B\",\"#EECA3B\",\"#B279A2\",\"#FF9DA6\",\"#9D755D\",\"#BAB0AC\"]")
```

Programming Project #1: Hybrid Images

CS445: Computational Photography

Part I: Hybrid Images

In [ ]:
```
```

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"cv2\",\"np\",\"LogNorm\",\"signal\",\"utils\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

import cv2

import numpy as np
from matplotlib.colors import LogNorm
from scipy import signal

# modify to where you store your project data including utils.py
# datadir = "/content/drive/My Drive/cs445_projects/proj1/"

# utilfn = datadir + "utils.py"
# !cp "$utilfn" .
import utils
```

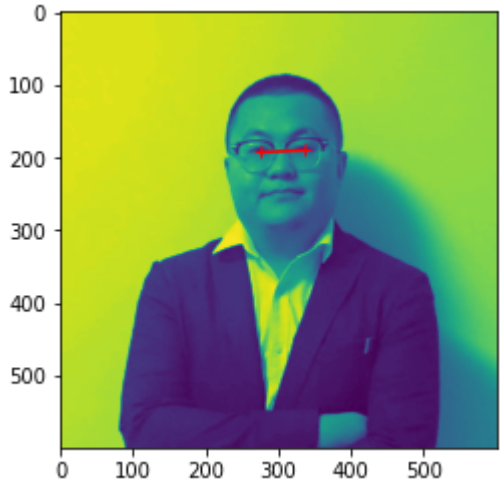
In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"plt\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

# switch from notebook to inline if using colab or otherwise cannot use interactive display)
import matplotlib.pyplot as plt
%matplotlib inline
```

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"im1_file\",\"im2_file\",\"im1\",\"im2\",\"np\",\"cv2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

im1_file = 'sun.jpg'
im2_file = 'cai.jpg'

im1 = np.float32(cv2.imread(im1_file, cv2.IMREAD_GRAYSCALE))
im2 = np.float32(cv2.imread(im2_file, cv2.IMREAD_GRAYSCALE))
```

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"pts_im1\",\"np\",\"plt\",\"im1\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

# pts_im1 = utils.prompt_eye_selection(im1)
# print(pts_im1)
pts_im1 = np.array([[275, 193], [335, 190]]) # uncomment if entering [x, y] pts manually
plt.plot(pts_im1[:,0], pts_im1[:,1], 'r-+')
plt.imshow(im1)
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7f67bf05c7c0>
```

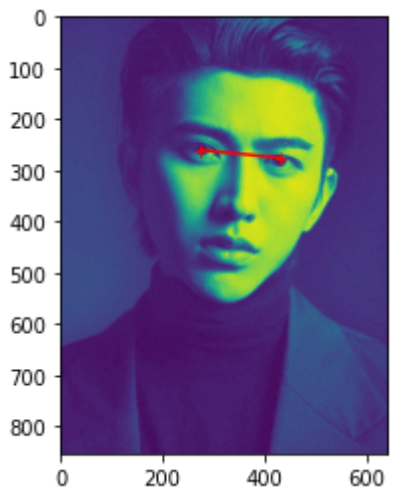
Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"pts_lady3\",\"pts_im2\",\"np\",\"plt\",\"im2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

pts_lady3 = np.array([[265, 210], [370, 230]])
# pts_im2 = utils.prompt_eye_selection(im2)
pts_im2 = np.array([[275,260], [430,275]]) # uncomment if entering [x, y] pts manually
plt.plot(pts_im2[:,0], pts_im2[:,1], 'r-+')
plt.imshow(im2)
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7f67bf76b520>
```

Out[ ]:

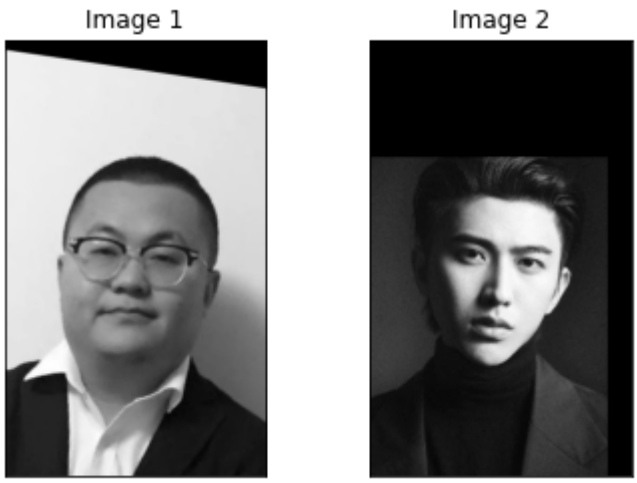


In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"im1\",\"im2\",\"utils\",\"im1_file\",\"im2_file\",\"pts_im1\",\"pts_im2\",\"cv2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

im1, im2 = utils.align_images(im1_file, im2_file, pts_im1, pts_im2, save_images=False)
im1 = cv2.cvtColor(im1, cv2.COLOR_BGR2GRAY) / 255.0
im2 = cv2.cvtColor(im2, cv2.COLOR_BGR2GRAY) / 255.0
```

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"plt\",\"im1\",\"im2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

#Images sanity check
fig, axes = plt.subplots(1, 2)
axes[0].imshow(im1,cmap='gray')
axes[0].set_title('Image 1'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(im2,cmap='gray')
axes[1].set_title('Image 2'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'Image 2'), [], [])
```

Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"hybridImage\",\"utils\",\"cv2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

def hybridImage(im1, im2, sigma_low, sigma_high):
    '''
    Inputs:
        im1:    RGB (height x width x 3) or a grayscale (height x width) image
                as a numpy array.
        im2:    RGB (height x width x 3) or a grayscale (height x width) image
                as a numpy array.
        sigma_low: standard deviation for the low-pass filter
        sigma_high: standard deviation for the high-pass filter

    Output:
        Return the combination of both images, one filtered with a low-pass filter
        and the other with a high-pass filter.
    '''
    low_filter = utils.gaussian_kernel(sigma_low, 25)
    high_filter = utils.gaussian_kernel(sigma_high, 25)

    low_pass_filtering = cv2.filter2D(im2, -1, low_filter)
    high_pass_filtering = cv2.filter2D(im1, -1, high_filter)

    high_pass_filtering = im1 - high_pass_filtering

    return low_pass_filtering, high_pass_filtering
```

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"sigma_low\",\"sigma_high\",\"low\",\"high\",\"im_hybrid\",\"hybridImage\",\"im1\",\"im2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

sigma_low = 5 # choose parameters that work for your images
sigma_high = 3
low, high = hybridImage(im1, im2, sigma_low, sigma_high)

im_hybrid= high * 0.3 + low * 0.7
```

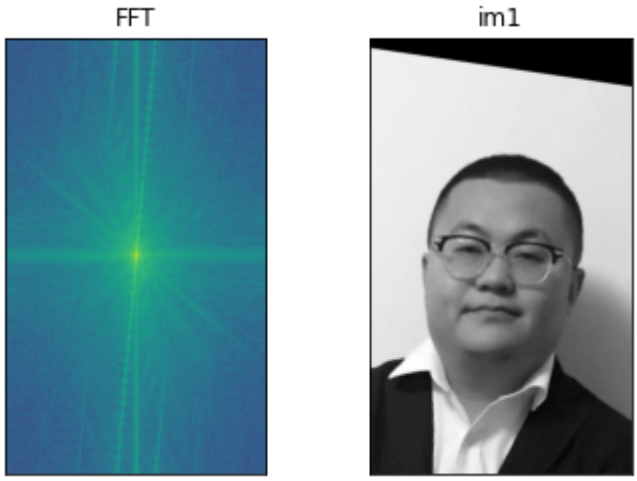
In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"cropped_object\",\"utils\",\"im_hybrid\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

# Optional: Select top left corner and bottom right corner to crop image
# the function returns dictionary of
# {
#   'cropped_image': np.ndarray of shape H x W
#   'crop_bound': np.ndarray of shape 2x2
# }
cropped_object = utils.interactive_crop(im_hybrid)
```

Out[ ]:

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"plt\",\"np\",\"im1\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
axes[0].imshow(np.log(np.abs(np.fft.fftshift(np.fft.fft2(im1)))))
axes[0].set_title('FFT'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(im1,cmap='gray')
axes[1].set_title('im1'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'im1'), [], [])
```

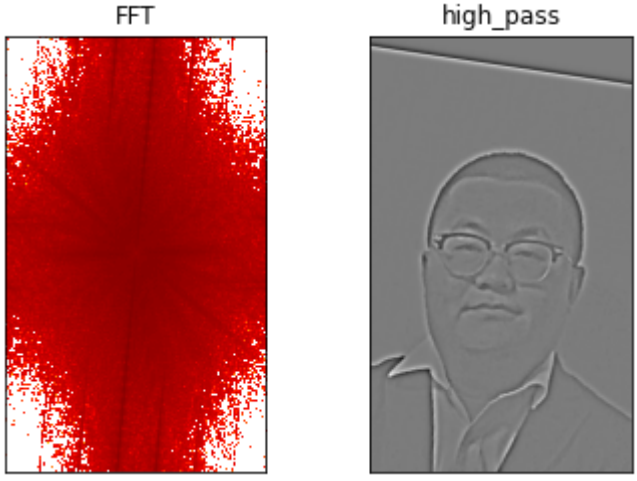
Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"fftmag\",\"plt\",\"np\",\"high\",\"LogNorm\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
fftmag = np.log(np.abs(np.fft.fftshift(np.fft.fft2(high))))
axes[0].imshow(fftmag,norm=LogNorm(fftmag.min(),fftmag.max()),cmap='jet')
axes[0].set_title('FFT'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(high,cmap='gray')
axes[1].set_title('high_pass'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'high_pass'), [], [])
```

Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"plt\",\"np\",\"im2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
axes[0].imshow(np.log(np.abs(np.fft.fftshift(np.fft.fft2(im2)))))
axes[0].set_title('FFT'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(im2,cmap='gray')
axes[1].set_title('im2'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

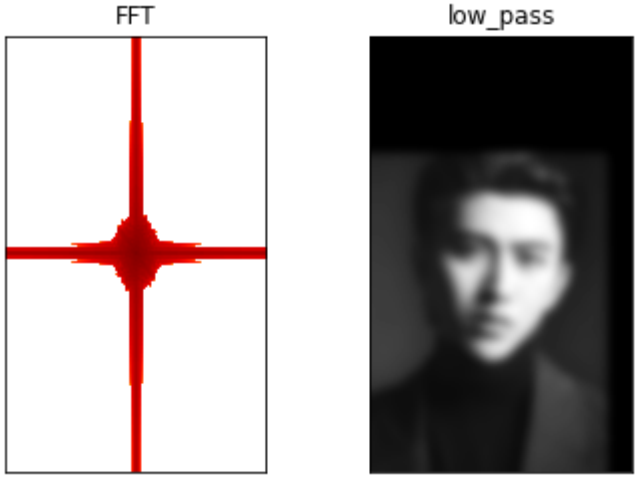
Out[ ]:
```
(Text(0.5, 1.0, 'im2'), [], [])
```

Out[ ]:

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"fftmag\",\"plt\",\"np\",\"low\",\"LogNorm\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
fftmag = np.log(np.abs(np.fft.fftshift(np.fft.fft2(low))))
axes[0].imshow(fftmag,norm=LogNorm(fftmag.min(),fftmag.max()),cmap='jet')
axes[0].set_title('FFT'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(low,cmap='gray')
axes[1].set_title('low_pass'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'low_pass'), [], [])
```

Out[ ]:

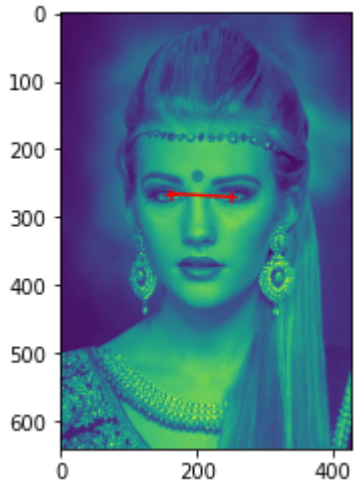


In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"lady_file\",\"man_file\",\"lady\",\"man\",\"np\",\"cv2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

lady_file = 'lady2_j.jpg'
man_file = 'man_j.jpg'
lady = np.float32(cv2.imread(lady_file, cv2.IMREAD_GRAYSCALE))
man = np.float32(cv2.imread(man_file, cv2.IMREAD_GRAYSCALE))
```

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"pts_lady\",\"np\",\"plt\",\"lady\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

pts_lady = np.array([[160, 265], [250, 270]]) # uncomment if entering [x, y] pts manually
plt.plot(pts_lady[:,0], pts_lady[:,1], 'r-+')
plt.imshow(lady)
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7f67c472aaf0>
```

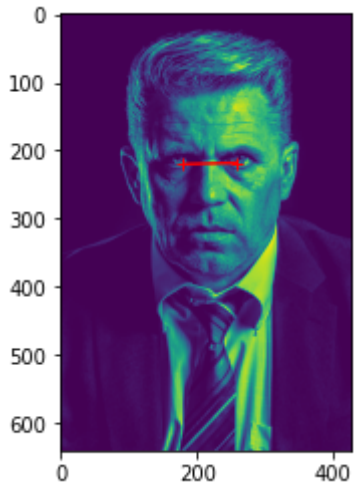
Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"pts_man\",\"np\",\"plt\",\"man\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

pts_man = np.array([[180, 220], [257, 218]]) # uncomment if entering [x, y] pts manually
plt.plot(pts_man[:,0], pts_man[:,1], 'r-+')
plt.imshow(man)
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7f67c6749f70>
```

Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"lady_im\",\"man_im\",\"utils\",\"lady_file\",\"man_file\",\"pts_lady\",\"pts_man\",\"cv2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

lady_im, man_im = utils.align_images(lady_file, man_file, pts_lady, pts_man, save_images=False)
lady_im = cv2.cvtColor(lady_im, cv2.COLOR_BGR2GRAY) / 255.0
man_im = cv2.cvtColor(man_im, cv2.COLOR_BGR2GRAY) / 255.0
```

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"plt\",\"lady_im\",\"man_im\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
axes[0].imshow(lady_im,cmap='gray')
axes[0].set_title('Image lady'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(man_im,cmap='gray')
axes[1].set_title('Image man'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'Image man'), [], [])
```

Out[ ]:

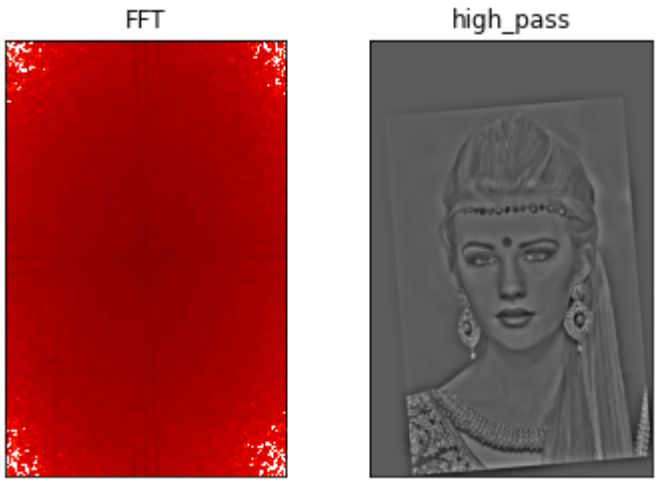
In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"sigma_low\",\"sigma_high\",\"low\",\"high\",\"im_hybrid_2\",\"cropped_object\",\"hybridImage\",\"lady_im\",\"man_im\",\"utils\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

sigma_low = 10 # choose parameters that work for your images
sigma_high = 8
low, high = hybridImage(lady_im, man_im, sigma_low, sigma_high)
im_hybrid_2= high * 0.3 + low * 0.7
cropped_object = utils.interactive_crop(im_hybrid_2)
```

Out[ ]:

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"fftmag\",\"plt\",\"np\",\"high\",\"LogNorm\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
fftmag = np.log(np.abs(np.fft.fftshift(np.fft.fft2(high))))
axes[0].imshow(fftmag,norm=LogNorm(fftmag.min(),fftmag.max()),cmap='jet')
axes[0].set_title('FFT'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(high,cmap='gray')
axes[1].set_title('high_pass'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'high_pass'), [], [])
```

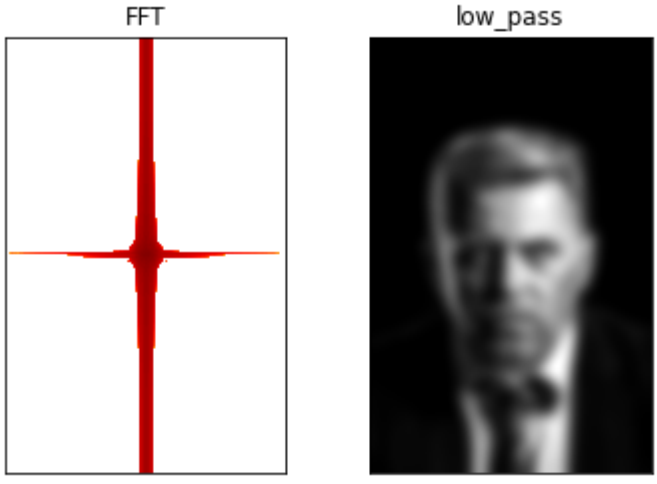
Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"fftmag\",\"plt\",\"np\",\"low\",\"LogNorm\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
fftmag = np.log(np.abs(np.fft.fftshift(np.fft.fft2(low))))
axes[0].imshow(fftmag,norm=LogNorm(fftmag.min(),fftmag.max()),cmap='jet')
axes[0].set_title('FFT'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(low,cmap='gray')
axes[1].set_title('low_pass'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'low_pass'), [], [])
```

Out[ ]:

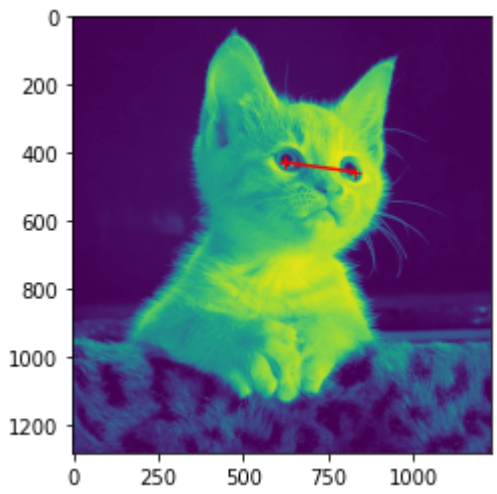


In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"cat_file\",\"lady3_file\",\"cat\",\"lady3\",\"np\",\"cv2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

cat_file = 'cat_j.jpg'
lady3_file = 'lady3.jpg'
cat = np.float32(cv2.imread(cat_file, cv2.IMREAD_GRAYSCALE))
lady3 = np.float32(cv2.imread(lady3_file, cv2.IMREAD_GRAYSCALE))
```

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"pts_cat\",\"np\",\"plt\",\"cat\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

pts_cat = np.array([[620, 427], [825, 455]])
plt.plot(pts_cat[:,0], pts_cat[:,1], 'r-+')
plt.imshow(cat)
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7f67c43df190>
```

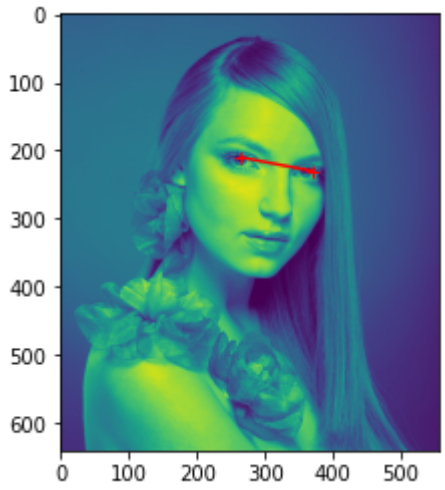
Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"pts_lady3\",\"np\",\"plt\",\"lady3\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

pts_lady3 = np.array([[265, 210], [370, 230]])
plt.plot(pts_lady3[:,0], pts_lady3[:,1], 'r-+')
plt.imshow(lady3)
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7f67c44b4c40>
```

Out[ ]:

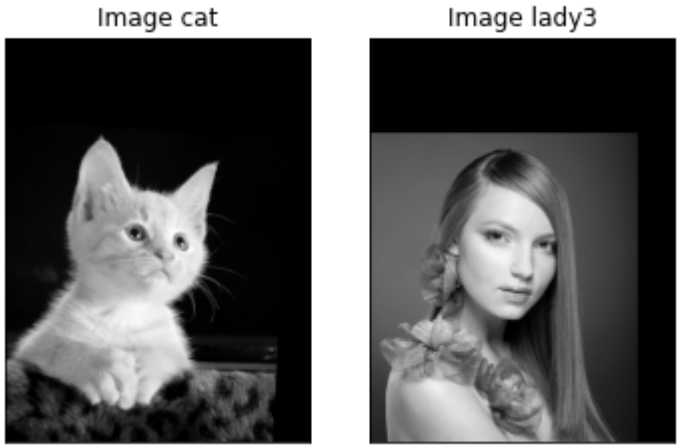


In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"cat\",\"lady3\",\"utils\",\"cat_file\",\"lady3_file\",\"pts_cat\",\"pts_lady3\",\"cv2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

cat, lady3 = utils.align_images(cat_file, lady3_file, pts_cat, pts_lady3, save_images=False)
cat = cv2.cvtColor(cat, cv2.COLOR_BGR2GRAY) / 255.0
lady3 = cv2.cvtColor(lady3, cv2.COLOR_BGR2GRAY) / 255.0
```

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"plt\",\"cat\",\"lady3\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
axes[0].imshow(cat,cmap='gray')
axes[0].set_title('Image cat'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(lady3,cmap='gray')
axes[1].set_title('Image lady3'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'Image lady3'), [], [])
```

Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"sigma_low\",\"sigma_high\",\"low\",\"high\",\"im_hybrid_3\",\"hybridImage\",\"cat\",\"lady3\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

sigma_low = 10 # choose parameters that work for your images
sigma_high = 8
low, high = hybridImage(cat, lady3, sigma_low, sigma_high)
im_hybrid_3= high * 0.3 + low * 0.7
```

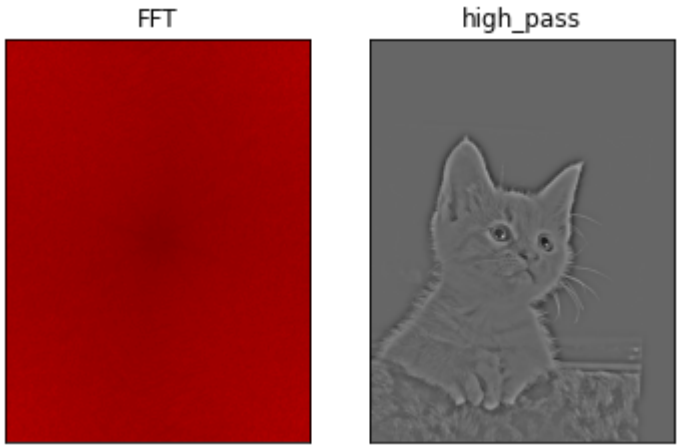
In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"cropped_object\",\"utils\",\"im_hybrid_3\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

cropped_object = utils.interactive_crop(im_hybrid_3)
```

Out[ ]:

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"fftmag\",\"plt\",\"np\",\"high\",\"LogNorm\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
fftmag = np.log(np.abs(np.fft.fftshift(np.fft.fft2(high))))
axes[0].imshow(fftmag,norm=LogNorm(fftmag.min(),fftmag.max()),cmap='jet')
axes[0].set_title('FFT'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(high,cmap='gray')
axes[1].set_title('high_pass'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'high_pass'), [], [])
```

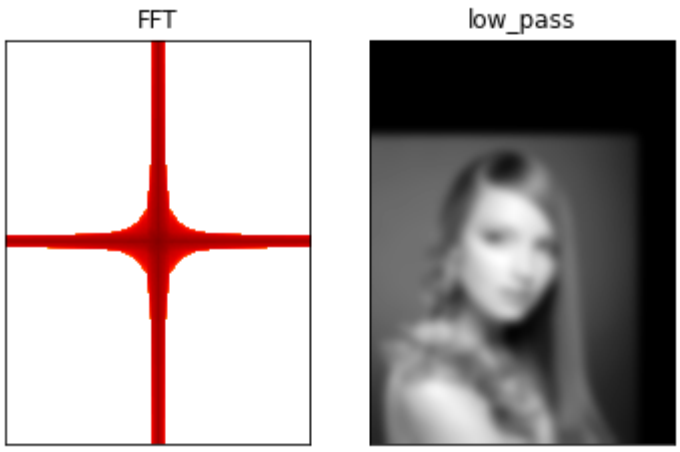
Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"fftmag\",\"plt\",\"np\",\"low\",\"LogNorm\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
fftmag = np.log(np.abs(np.fft.fftshift(np.fft.fft2(low))))
axes[0].imshow(fftmag,norm=LogNorm(fftmag.min(),fftmag.max()),cmap='jet')
axes[0].set_title('FFT'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(low,cmap='gray')
axes[1].set_title('low_pass'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'low_pass'), [], [])
```

Out[ ]:

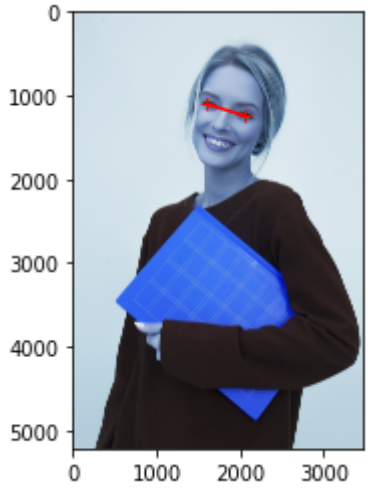


In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"image_3_file\",\"image_4_file\",\"min_width\",\"min_height\",\"cv2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

image_3_file = cv2.imread('resized_girl.jpg')
image_4_file = cv2.imread('resized_man.jpg')

min_width = min(image_3_file.shape[1], image_4_file.shape[1])
min_height = min(image_3_file.shape[0], image_4_file.shape[0])

image_3_file = cv2.resize(image_3_file, (min_width, min_height))
image_4_file = cv2.resize(image_4_file, (min_width, min_height))
```

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"pts_im3\",\"image_3_file\",\"np\",\"plt\",\"cv2\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

pts_im3 = np.array([[1600, 1100], [2050, 1240]]) # uncomment if entering [x, y] pts manually
plt.plot(pts_im3[:,0], pts_im3[:,1], 'r-+')
image_3_file = cv2.imread('resized_girl.jpg')
plt.imshow(image_3_file)
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7f67bf26bc10>
```

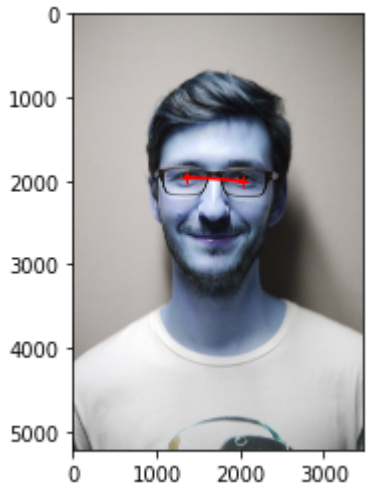
Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"pts_im4\",\"np\",\"plt\",\"image_4_file\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

pts_im4 = np.array([[1350, 1950], [2040, 2000]]) # uncomment if entering [x, y] pts manually
plt.plot(pts_im4[:,0], pts_im4[:,1], 'r-+')
plt.imshow(image_4_file)
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7f67bfc88c40>
```

Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"image_3\",\"image_4\",\"fig\",\"axes\",\"cv2\",\"image_3_file\",\"image_4_file\",\"utils\",\"pts_im3\",\"pts_im4\",\"plt\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

cv2.imwrite('resized_girl.jpg', image_3_file)
cv2.imwrite('resized_man.jpg', image_4_file)

image_3, image_4 = utils.align_images('resized_girl.jpg', 'resized_man.jpg',pts_im3,pts_im4,save_images=False)
image_3 = cv2.cvtColor(image_3, cv2.COLOR_BGR2RGB)
image_4 = cv2.cvtColor(image_4, cv2.COLOR_BGR2RGB)

fig, axes = plt.subplots(1, 2)
axes[0].imshow(image_3,cmap='gray')
axes[0].set_title('Image 3'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(image_4,cmap='gray')
axes[1].set_title('Image 4'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

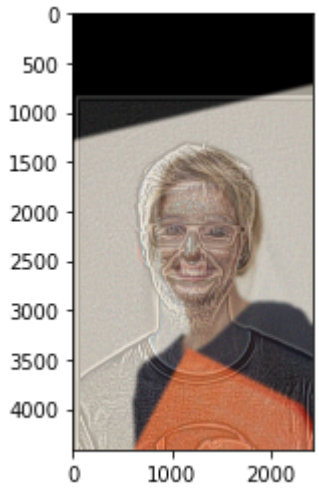
Out[ ]:
```
(Text(0.5, 1.0, 'Image 4'), [], [])
```

Out[ ]:

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"sigma_low\",\"sigma_high\",\"low_im\",\"high_im\",\"im_hybrid\",\"image_3\",\"image_4\",\"hybridImage\",\"plt\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))


sigma_low = 50 # choose parameters that work for your images
sigma_high = 100

low_im = image_3.copy()
high_im = image_4.copy()

low_im[:,:,0], high_im[:,:,0] = hybridImage(high_im[:,:,0], low_im[:,:,0], sigma_low, sigma_high)
low_im[:,:,1], high_im[:,:,1] = hybridImage(high_im[:,:,1], low_im[:,:,1], sigma_low, sigma_high)
low_im[:,:,2], high_im[:,:,2] = hybridImage(high_im[:,:,2], low_im[:,:,2], sigma_low, sigma_high)

im_hybrid = low_im.copy()
im_hybrid[:,:,0] = low_im[:,:,0] * 0.7 + 0.3 * high_im[:,:,0]
im_hybrid[:,:,1] = low_im[:,:,1] * 0.7 + 0.3 * high_im[:,:,1]
im_hybrid[:,:,2] = low_im[:,:,2] * 0.7 + 0.3 * high_im[:,:,2]

plt.imshow(im_hybrid)
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7f67baad67c0>
```

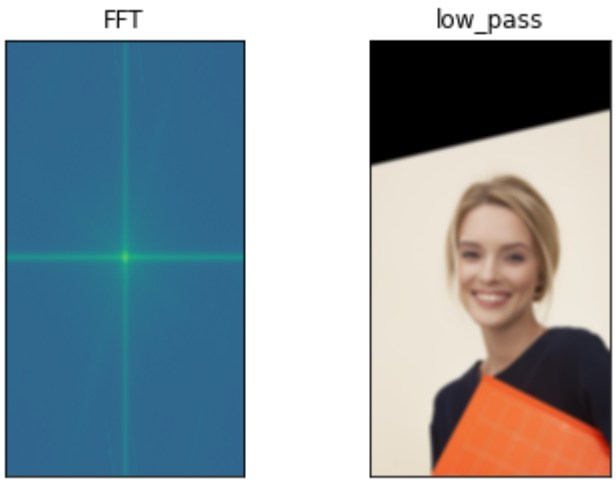
Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"plt\",\"np\",\"cv2\",\"low_im\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
axes[0].imshow(np.log(np.abs(np.fft.fftshift(np.fft.fft2(cv2.cvtColor(low_im, cv2.COLOR_BGR2GRAY))))))
axes[0].set_title('FFT'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(low_im)
axes[1].set_title('low_pass'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'low_pass'), [], [])
```

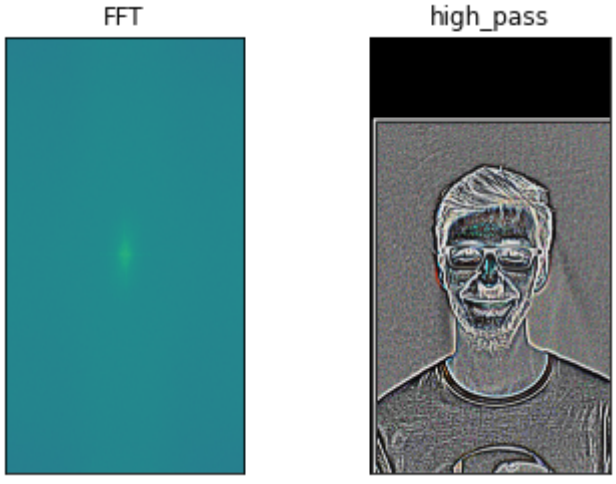
Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"fig\",\"axes\",\"plt\",\"np\",\"cv2\",\"high_im\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

fig, axes = plt.subplots(1, 2)
axes[0].imshow(np.log(np.abs(np.fft.fftshift(np.fft.fft2(cv2.cvtColor(high_im, cv2.COLOR_BGR2GRAY))))))
axes[0].set_title('FFT'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(high_im)
axes[1].set_title('high_pass'), axes[1].set_xticks([]), axes[1].set_yticks([])
```

Out[ ]:
```
(Text(0.5, 1.0, 'high_pass'), [], [])
```

Out[ ]:



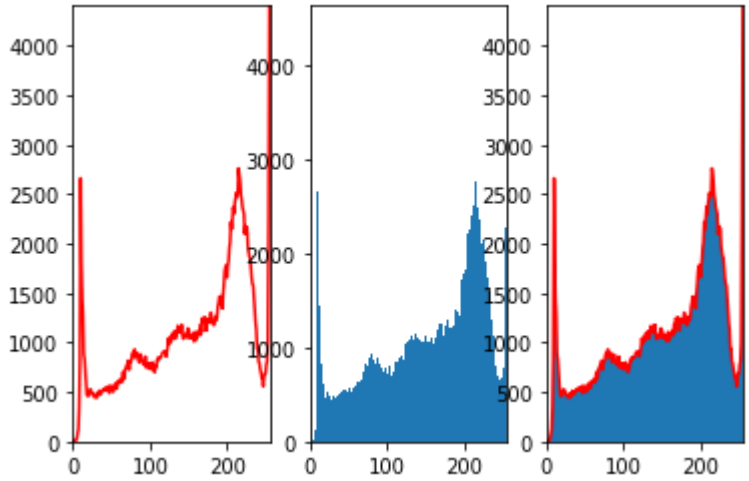
Part II: Image Enhancement

Two out of three types of image enhancement are required. Choose a good image to showcase each type and implement a method. This code doesn't rely on the hybrid image part.

Contrast enhancement

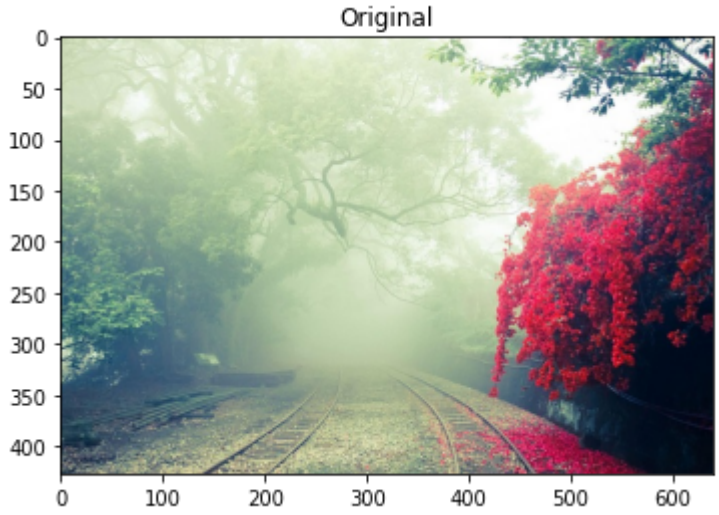
In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"contrast\",\"hist_size\",\"hist\",\"cv2\",\"plt\",\"np\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

contrast = cv2.imread('contrast.jpg', cv2.IMREAD_GRAYSCALE)
# print(contrast.shape)
# min_value = contrast.min()
# max_value = contrast.max()
# print(min_value)
# print(max_value)
hist_size = [256]
hist = cv2.calcHist([contrast],[0], None, hist_size, [0, 256])

plt.subplot(1,3,1),plt.plot(hist, color='r'),plt.axis([0,256,0,np.max(hist)])

plt.subplot(1,3,2),plt.hist(contrast.ravel(),bins=256,range=[0,256]),plt.xlim([0,256])

plt.subplot(1,3,3), plt.plot(hist,color="r"),plt.axis([0,256,0,np.max(hist)])
plt.hist(contrast.ravel(),bins=256,range=[0,256]),plt.xlim([0,256])

plt.show()
```

Out[ ]:

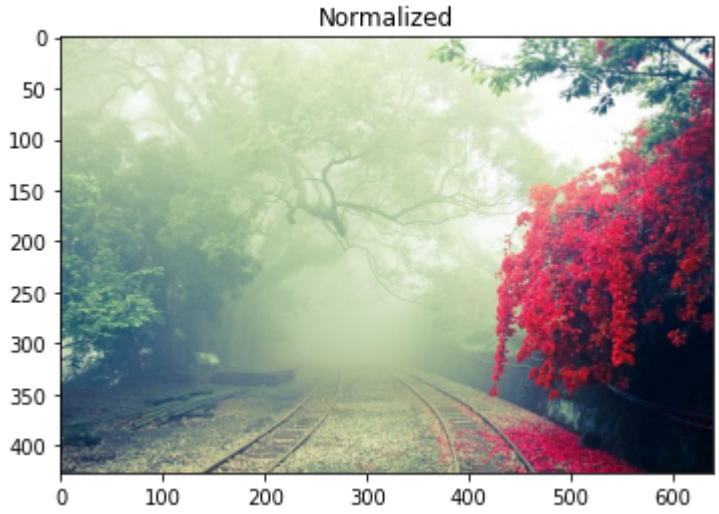


In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"image\",\"image_norm\",\"cv2\",\"plt\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

## this code just try for the normalization
image = cv2.imread('contrast.jpg')
# max_value = image.max()
# print(max_value)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
image_norm = cv2.normalize(image, dst=None, alpha=0, beta=255, norm_type=cv2.NORM_MINMAX)
plt.imshow(image), plt.title('Original')
plt.show()
plt.imshow(image_norm), plt.title('Normalized')
plt.show()
```

Out[ ]:



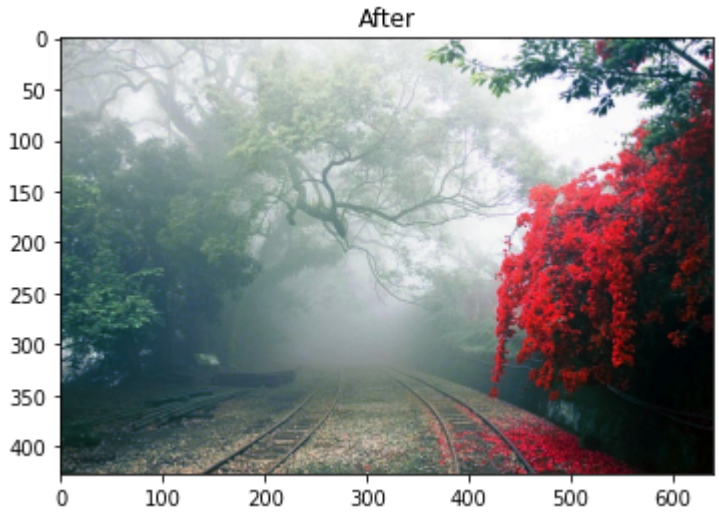
Out[ ]:



In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"image\",\"red\",\"green\",\"blue\",\"red_equal\",\"green_equal\",\"blue_equal\",\"image_equal\",\"cv2\",\"plt\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

## this code works for contrast enhancement
image = cv2.imread('contrast.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
red,green,blue = cv2.split(image)
red_equal = cv2.equalizeHist(red)
green_equal = cv2.equalizeHist(green)
blue_equal = cv2.equalizeHist(blue)
image_equal = cv2.merge((red_equal, green_equal, blue_equal))
plt.imshow(image),plt.title("Original")
plt.show()
plt.imshow(image_equal),plt.title("After")
plt.show()
```

Out[ ]:

Out[ ]:



Color enhancement

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"hist_ave_2\",\"color\",\"color_\",\"cv2\",\"np\",\"plt\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

## method 1
def hist_ave_2(src):
    src[:,:,1] = cv2.equalizeHist(src[:,:,1])
    src[:,:,1] = np.clip(src[:,:,1], 0, 255)
    return src

color = cv2.imread("colorshift.jpg")
color = cv2.cvtColor(color, cv2.COLOR_BGR2RGB)

color_ = cv2.imread("colorshift.jpg")
color_ = cv2.cvtColor(color_, cv2.COLOR_BGR2HSV)
color_ = hist_ave_2(color_)
color_=cv2.cvtColor(color_,cv2.COLOR_HSV2RGB)

plt.imshow(color),plt.title("Original")
plt.show()

plt.imshow(color_),plt.title("Color Enhancement")
plt.show()
```

Out[ ]:

Out[ ]:

In [ ]:
```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"color\",\"color_\",\"cv2\",\"plt\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

## method 2
color = cv2.imread("colorshift.jpg")
color = cv2.cvtColor(color, cv2.COLOR_BGR2RGB)

color_ = cv2.imread("colorshift.jpg")
color_ = cv2.cvtColor(color_, cv2.COLOR_BGR2HSV)

color_[:,:,1] = color_[:,:,1] * 2
color_[:,:,1][color_[:,:,1] > 255] = 255
color_=cv2.cvtColor(color_,cv2.COLOR_HSV2RGB)

plt.imshow(color),plt.title("Original")
plt.show()

plt.imshow(color_),plt.title("Color Enhancement")
plt.show()
```

Out[ ]:

Out[ ]:

Color shift

In [ ]:

```
import json as _hex_json
_hex_pks.kernel_execution.scope_watcher.mark_dirty_scope_items(args=_hex_types.MarkDirtyScopeItemsArgs.from_dict({**_hex_json.loads("{\"dirty_scope_items\":[\"image\",\"more_red\",\"less_yellow\",\"cv2\",\"np\",\"plt\"]}")}), app_session_token=_hex_APP_SESSION_TOKEN, python_kernel_init_status=_hex_python_kernel_init_status, hex_timezone=_hex_kernel.variable_or_none("hex_timezone", scope_getter=lambda: globals()))

image = cv2.imread('temp_j.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2Lab)
print(image[:,:,1].min())

more_red = image.copy()
more_red[:,:,1] = more_red[:,:,1] + 20
more_red = np.clip(more_red, 0, 255)

less_yellow = image.copy()
less_yellow[:,:,2] = less_yellow[:,:,2] - 20
less_yellow = np.clip(less_yellow, 0, 255)

more_red = cv2.cvtColor(more_red, cv2.COLOR_Lab2RGB)
less_yellow = cv2.cvtColor(less_yellow, cv2.COLOR_Lab2RGB)
image = cv2.cvtColor(image, cv2.COLOR_Lab2RGB)

plt.imshow(image),plt.title("Original")
plt.show()

plt.imshow(more_red),plt.title("More Red")
plt.show()

plt.imshow(less_yellow),plt.title("Less Yellow")
plt.show()
```

Out[ ]: 76

Out[ ]:

Out[ ]:

Out[ ]: