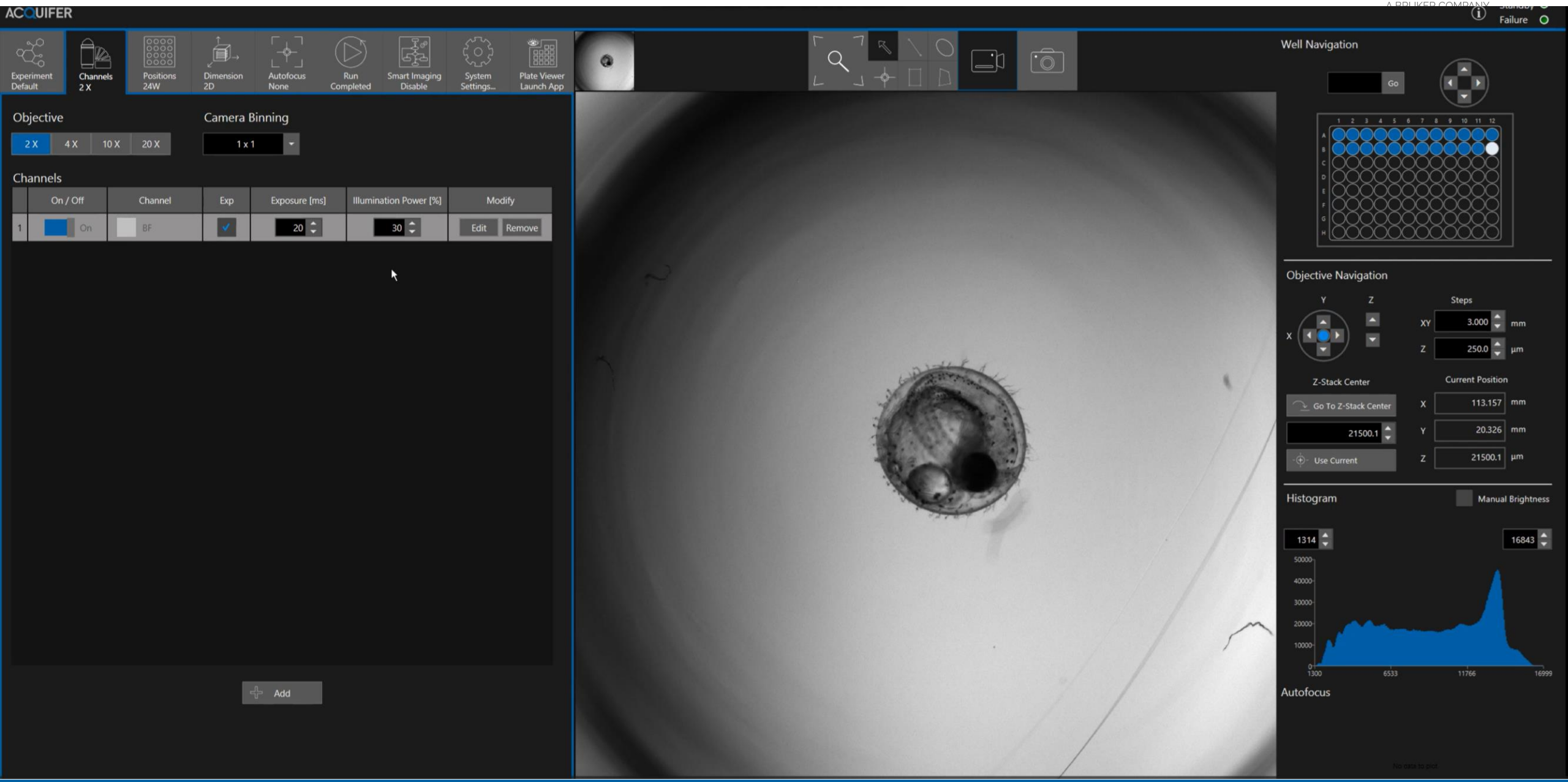
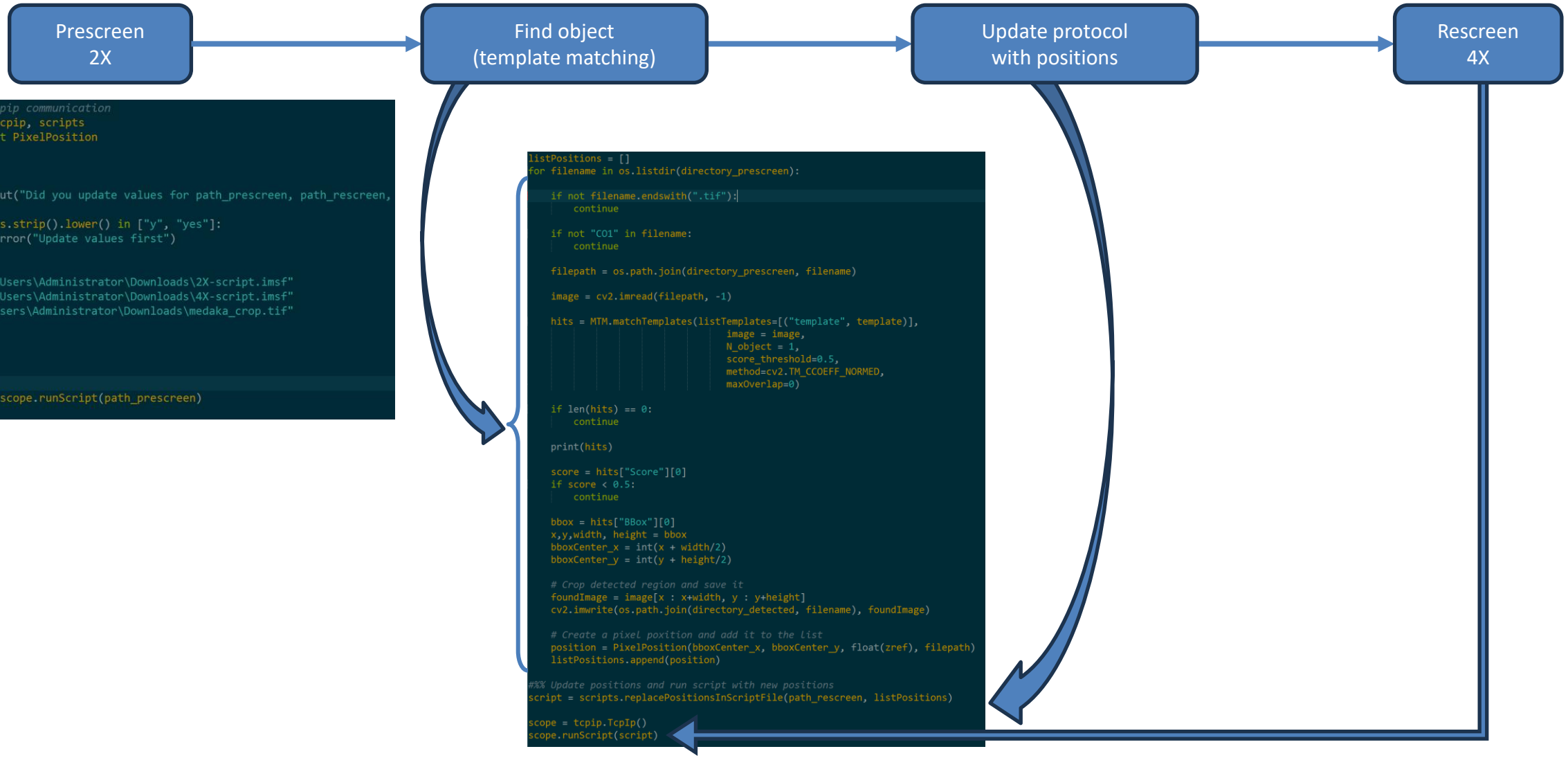
Prescreen/Rescreen workflow on an ACQUIFER Imaging Machine

ACQUIFER
A BRUKER COMPANY

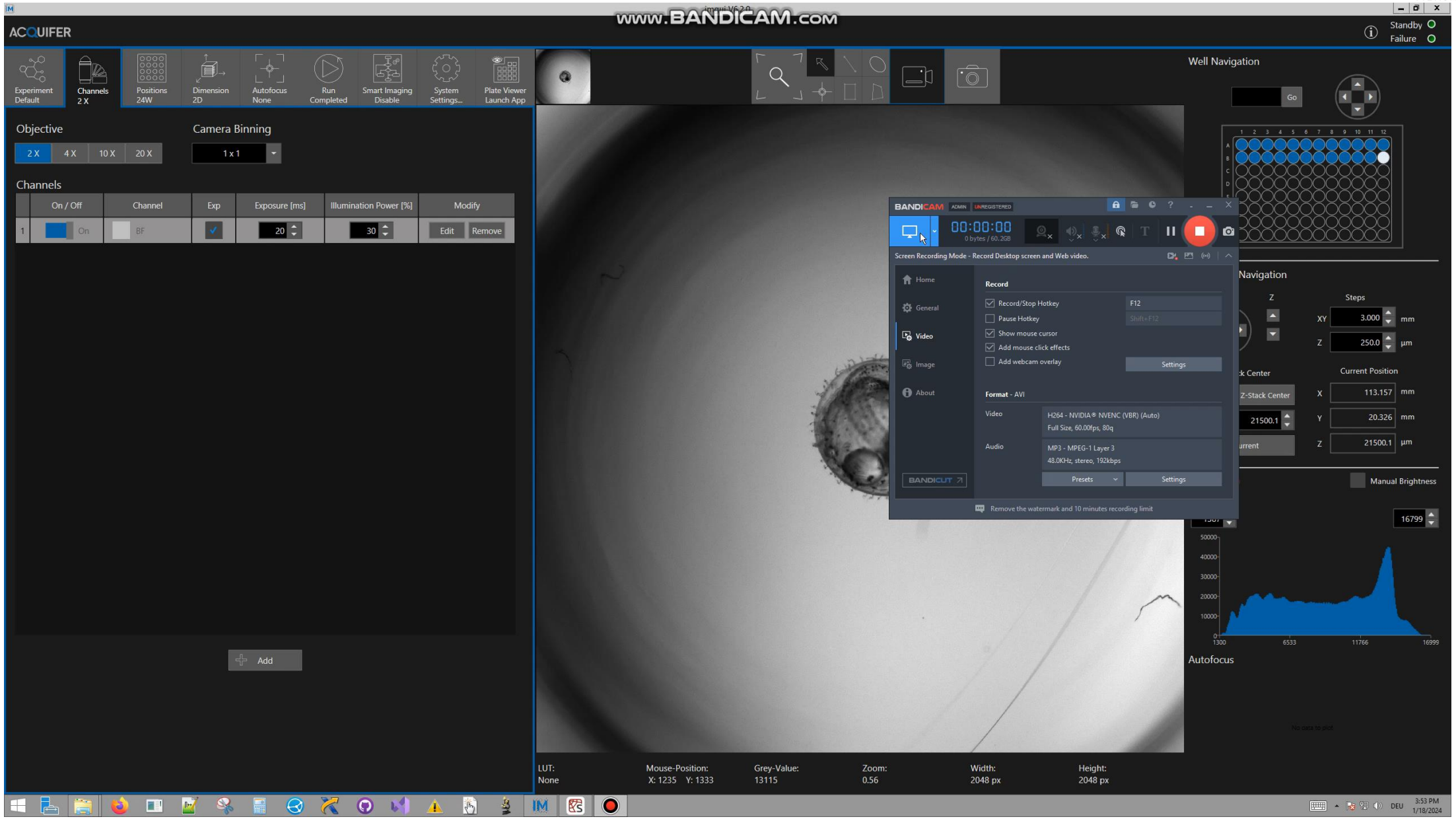

ACQUIFER
Failure
Experiment Default
Channels 2 X
Positions 24W
Dimension 2D
Autofocus None
Run Completed
Smart Imaging Disable
System Settings...
Plate Viewer Launch App
Objective
2 X
4 X
10 X
20 X
Camera Binning
1 x 1
Channels
On / Off
Channel
Exp
Exposure [ms]
Illumination Power [%]
Modify
1
On
BF
20
30
Edit
Remove
Add
Well Navigation
Go
Objective Navigation
Y
Z
X
Steps
XY
3.000
mm
Z
250.0
µm
Z-Stack Center
Go To Z-Stack Center
21500.1
Use Current
Current Position
X
113.157
mm
Y
20.326
mm
Z
21500.1
µm
Histogram
Manual Brightness
1314
16843
50000
40000
30000
20000
10000
0
1300
6533
11766
16999
Autofocus

Sequential Prescreen/Rescreen workflow - python




```python
#%% Import and open tcpip communication
from acquifer import tcpip, scripts
from ScriptUtils import PixelPosition
import MTM, cv2, os

#%% Checks
hasUpdatedValues = input("Did you update values for path_prescreen, path_rescreen,

if not hasUpdatedValues.strip().lower() in ["y", "yes"]:
    raise InterruptedError("Update values first")

#%% Input scripts
path_prescreen = r"C:\Users\Administrator\Downloads\2X-script.imsf"
path_rescreen  = r"C:\Users\Administrator\Downloads\4X-script.imsf"
path_template = r"C:\Users\Administrator\Downloads\medaka_crop.tif"

zref = 21500.1

#%% Run prescript
scope = tcpip.TcpIp()
directory_prescreen = scope.runScript(path_prescreen)
```

```python
listPositions = []
for filename in os.listdir(directory_prescreen):

    if not filename.endswith(".tif"):
        continue

    if not "C01" in filename:
        continue

    filepath = os.path.join(directory_prescreen, filename)

    image = cv2.imread(filepath, -1)

    hits = MTM.matchTemplates(listTemplates=[("template", template)],
                              image = image,
                              N_object = 1,
                              score_threshold=0.5,
                              method=cv2.TM_CCOEFF_NORMED,
                              maxOverlap=0)

    if len(hits) == 0:
        continue

    print(hits)

    score = hits["Score"][0]
    if score < 0.5:
        continue

    bbox = hits["BBox"][0]
    x,y,width, height = bbox
    bboxCenter_x = int(x + width/2)
    bboxCenter_y = int(y + height/2)

    # Crop detected region and save it
    foundImage = image[x : x+width, y : y+height]
    cv2.imwrite(os.path.join(directory_detected, filename), foundImage)

    # Create a pixel poxition and add it to the list
    position = PixelPosition(bboxCenter_x, bboxCenter_y, float(zref), filepath)
    listPositions.append(position)

#%% Update positions and run script with new positions
script = scripts.replacePositionsInScriptFile(path_rescreen, listPositions)

scope = tcpip.TcpIp()
scope.runScript(script)
```


www.BANDICAM.com
ACQUIFER
Standby
Failure
Experiment Default
Channels 2 X
Positions 24W
Dimension 2D
Autofocus None
Run Completed
Smart Imaging Disable
System Settings...
Plate Viewer Launch App
Objective
2 X
4 X
10 X
20 X
Camera Binning
1 x 1
Channels
On / Off
Channel
Exp
Exposure [ms]
Illumination Power [%]
Modify
1
On
BF
20
30
Edit
Remove
Add
Well Navigation
Go
Navigation
Steps
XY
3.000
mm
Z
250.0
µm
Current Position
X
113.157
mm
Y
20.326
mm
Z
21500.1
µm
Manual Brightness
16799
Autofocus
BANDICAM
00:00:00
Screen Recording Mode - Record Desktop screen and Web video.
Home
General
Video
Image
About
Record
Record/Stop Hotkey
F12
Pause Hotkey
Shift+F12
Show mouse cursor
Add mouse click effects
Add webcam overlay
Settings
Format - AVI
Video
H264 - NVIDIA® NVENC (VBR) (Auto)
Full Size, 60.00fps, 80q
Audio
MP3 - MPEG-1 Layer 3
48.0KHz, stereo, 192kbps
Presets
Settings
BANDICUT
Remove the watermark and 10 minutes recording limit
LUT:
None
Mouse-Position:
X: 1235 Y: 1333
Grey-Value:
13115
Zoom:
0.56
Width:
2048 px
Height:
2048 px
DEU
3:53 PM
1/18/2024

Video at : …

Pros

- simple / high-level

- single short python file (+ protocol for control software)

Cons

- Sequential : no „real time“ feedback

- No direct access to the microscope parameters (exposure, light-source intensity…)

Note : „low-level“/fine grain approach possible too but more complex/verbose

ACQUIFER Imaging Machine

Various options for smart-microscopy

Imaging Machine (IM)

IM Control Software

C#-internal scripting

External applications
python, java....

Plate-Viewer plugins