ICEA

UNIVERSITÀ DEGLI STUDI DI PADOVA

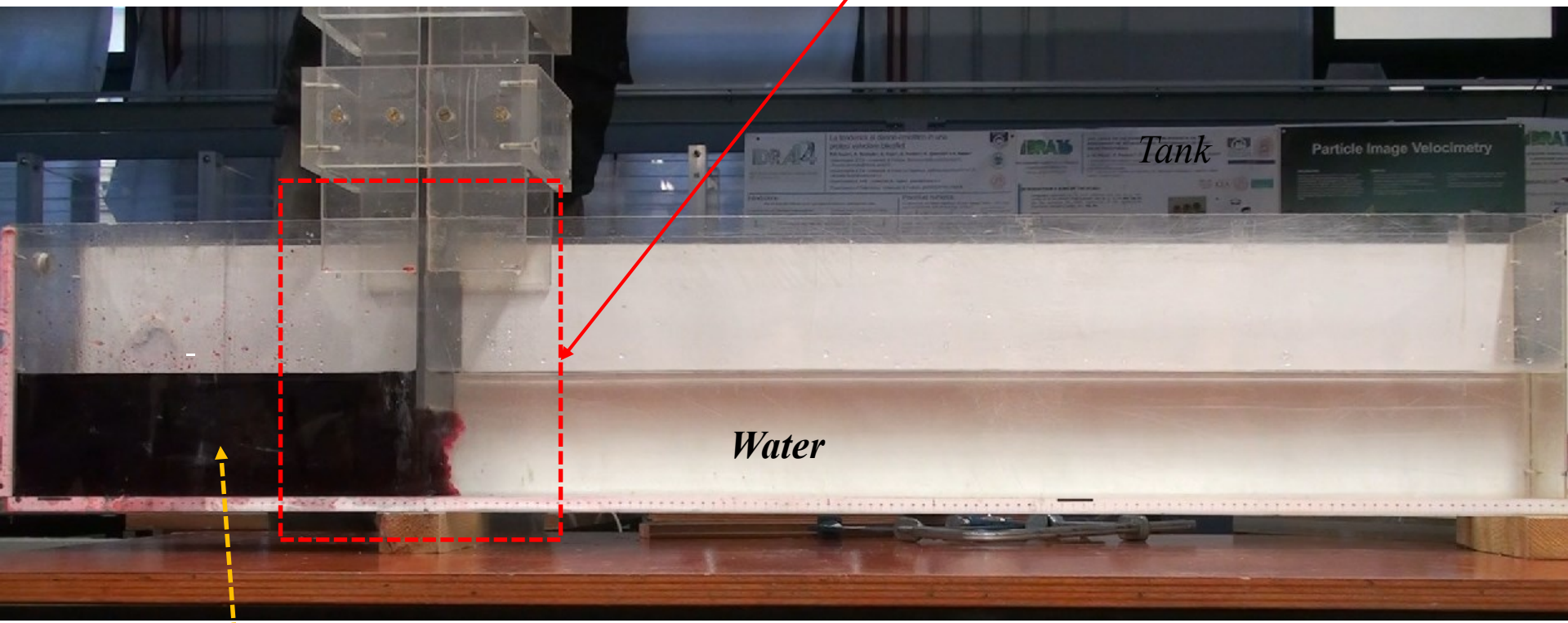
Lesson 4:
MATLAB: Lock Exchange

2020

Luigi Di Micco, Enrica Belluco &
Paolo Peruzzo

luigi.dimicco@dicea.unipd.it

Experimental Set-Up:



$\rho_{(Water + salt + red\ dye)} \approx$ ***from 1005 to 1020 kg/m^3***

i)

```
%% Lock Exchange Video Elaboration

% Initialization and creation of the working folder
clear all;
close all;
clc
imtool close all;  % Close all imtool figures if you have the Image Processing Toolbox.
workspace;  % Make sure the workspace panel is showing.

% Creation of the working folder
mkdir estratte_dens1008_1
Folder = 'estratte_dens1008_1';
```

ii)

```
%% Load the video (.avi) for the analysis
prompt = {'Please enter the .avi video name:'};
dlg_title = 'Input';
num_lines = 1;
defaultans = {'100x_x.avi'};
answer = inputdlg(prompt,dlg_title,num_lines,defaultans);
video_rsz=answer{1};
tic;
```

iii)

```
%% Digit the fps (frame per second) and define the initial an
fps=25;
iSTART=7*fps; % t_start(s)*25(frames/s)
iSTOP=26*fps;  % t_end(s)*25(frames/s)
vid1=VideoReader('1008_1.mp4');
```

i) Creation of a new Folder "*estratte_densità10xx_x*", inside which you will find all processed images and saved files;

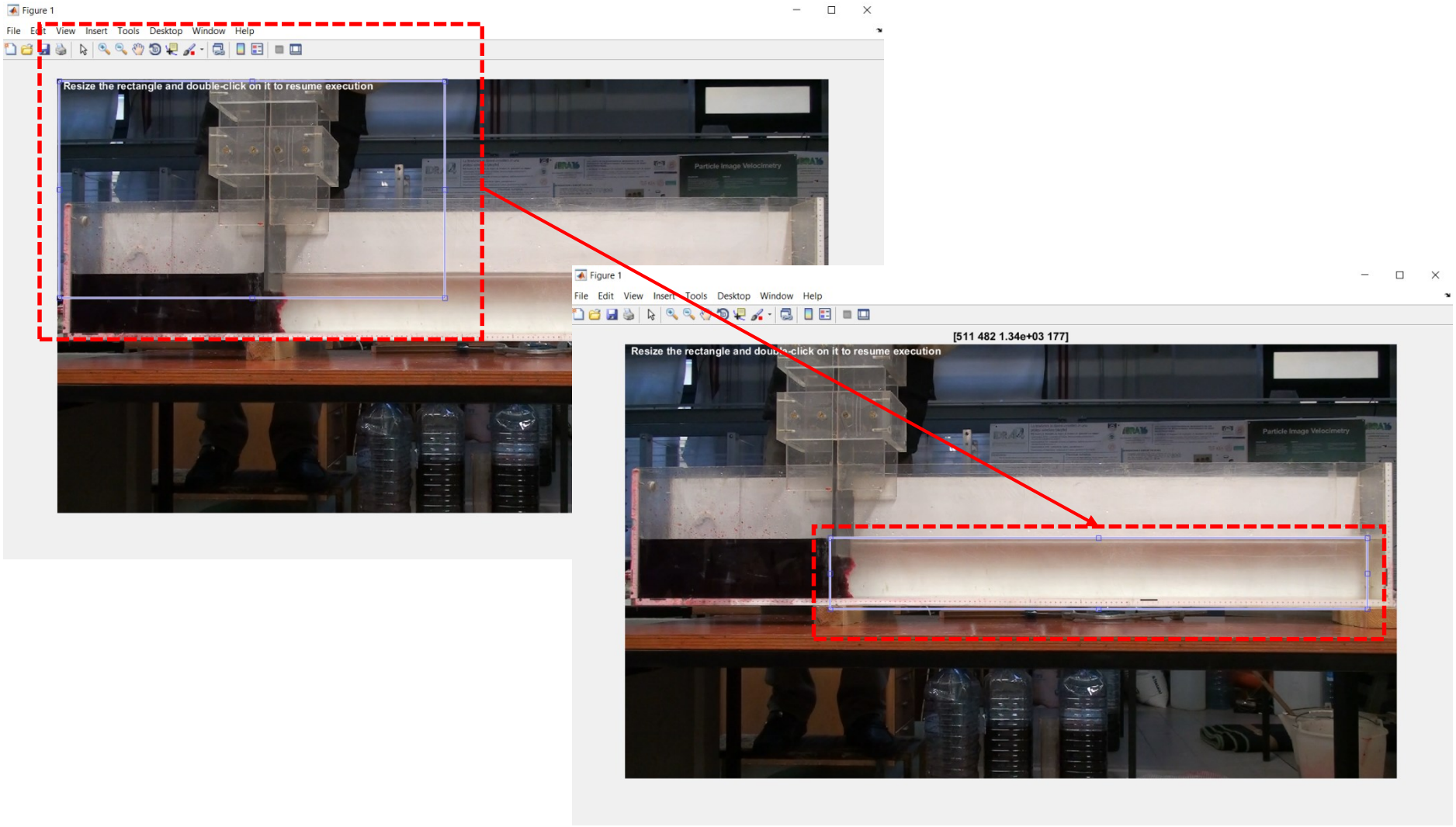
ii) Loading of the video;

iii) Definition of the frames of interest for the analysis.

```
%% RESIZE
n=vid1.NumberOfFrames;
writerObj1 = VideoWriter(video_rsz);
open(writerObj1);
im=read(vid1,iSTART);
% figure, imshow(im);
% Double-click on the rectangle to resume execution of the MATLAB command line
[pos] = getRoi(im);

%% New video defined in the investigation area according to the "pos" definition
for iFrame = iSTART:iSTOP

    im=read(vid1,iFrame);
    %   im=imresize(im,0.5);
    imc=imcrop(im,[pos]);% The dimention of the new video
    %   img=rgb2gray(im);
    %   [a,b]=size(img);
    %   imc=imresize(imc,[a,b]);
    writeVideo(writerObj1,imc);
    %     subplot(2,1,1)
    %     imshow(im)
    %     subplot(2,1,2)
    %     imshow(imc)
end
close(writerObj1)
imwrite(imc, 'calibrazione_1008_1.jpg');
% imwrite(imc, fullfile(Folder, 'calibrazione.jpg'));
```

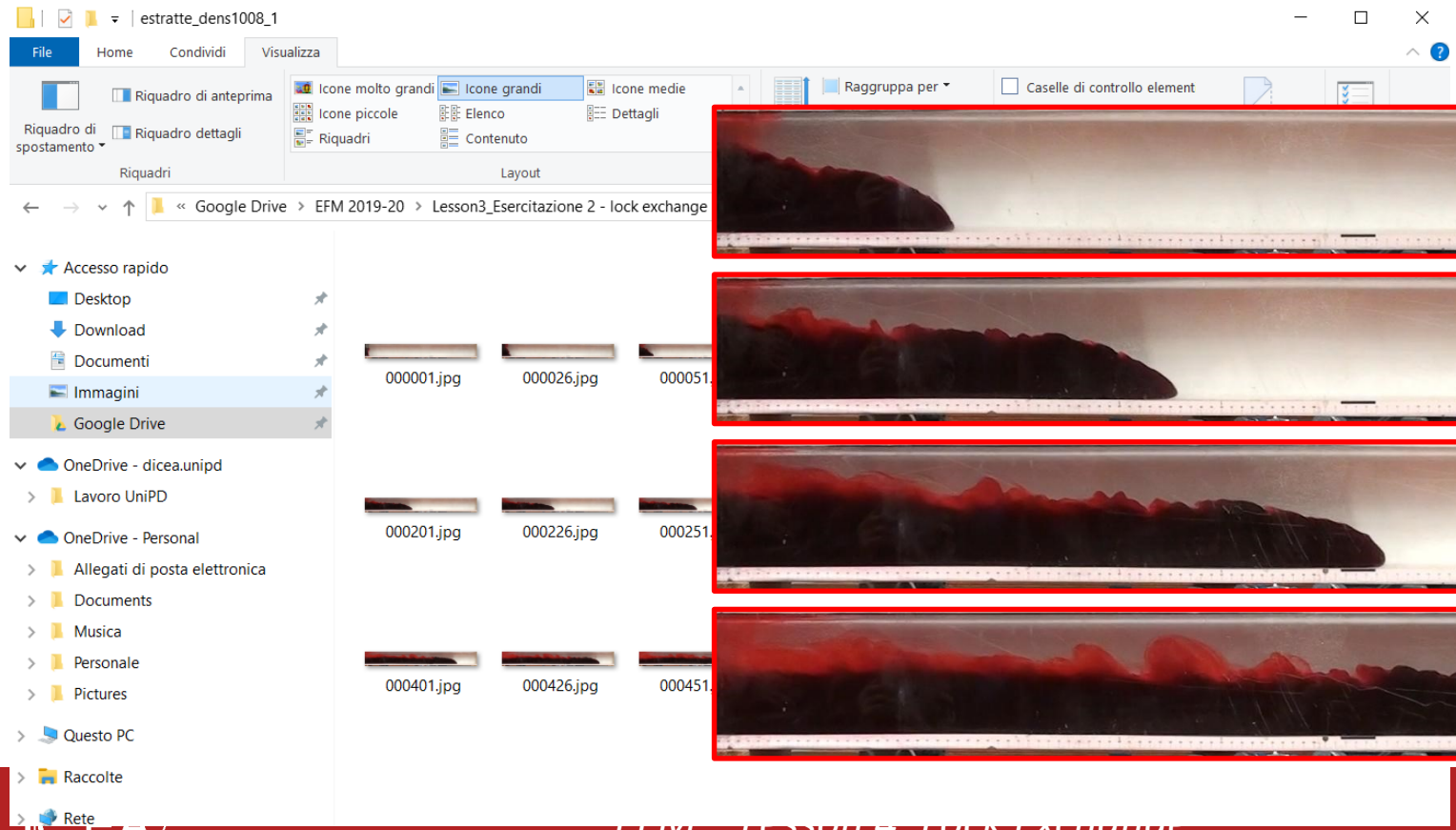
Definition of a clipping rectangular = area of interest of your analysis.

Definition of a clipping rectangular = area of interest of your analysis.



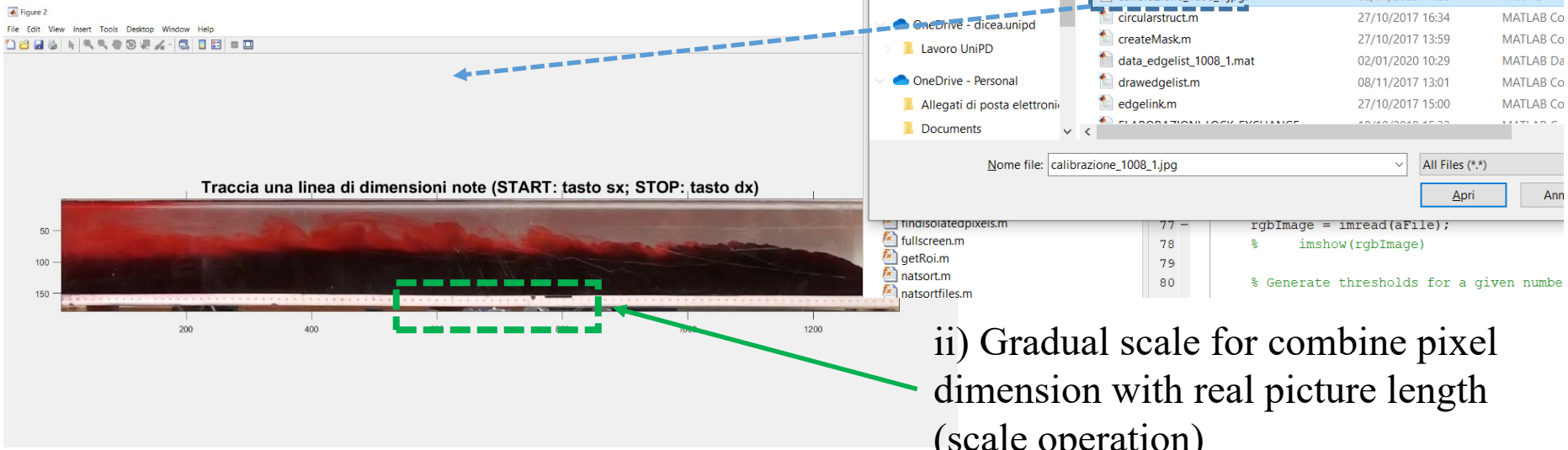
MatLab Code: Analysis

```
%% Extraction of the images from the video (each time_gap; to define)

vid=VideoReader(video_rsz);
n=vid.NumberOfFrames;
time_gap=25;
for iFrame = 1:time_gap:n
    frames = read(vid, iFrame);
    imwrite(frames, fullfile(Folder, sprintf('%06d.jpg', iFrame)));
end
```



```
22
23      %% Size CALIBRATION for obtain pixel size
23      % Definition of a line of known dimensions on the image
24 -    [dim_pixel]=spatial_calibration_EB()
25
```

i) Select the figure calibrazione 100xx_x from the list below

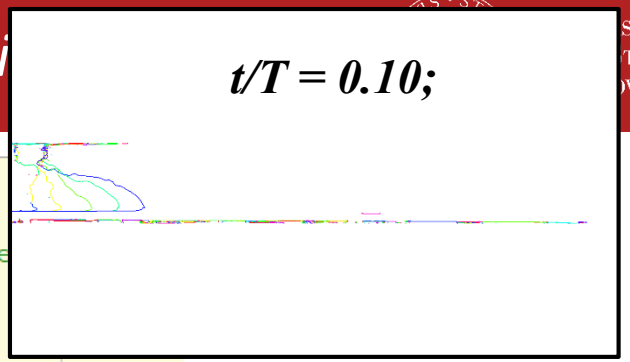


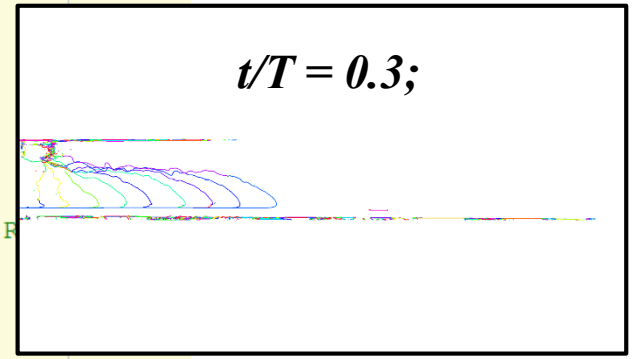
ii) Gradual scale for combine pixel dimension with real picture length (scale operation)

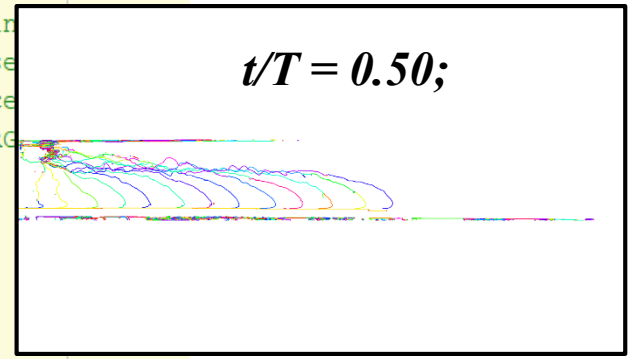
MatLab Code: Analysi

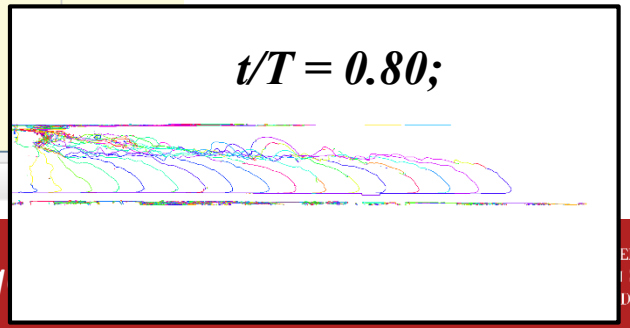
```
%% EDGE DETECTION
% h = waitbar(0,'Running......');
  FileList = dir(fullfile(Folder, '*.jpg')); % get list of files in dire
  [~,ndx] = natsortfiles({FileList.name}); % indices of correct order
  FileList = FileList(ndx); % sort structure using indices

for iFile = 1:length(FileList)
    aFile = fullfile(Folder, FileList(iFile).name);
    rgbImage = imread(aFile);
    %     imshow(rgbImage)

    % Generate thresholds for a given number of levels from the entire R

     [quantPlane]=Threshold_EB(rgbImage,7);
%     imshow(quantPlane)
%     %  [BW,MASKEDRGBIMAGE] = createMask(RGB) thresholds image RGB usin
    %  minimum/maximum values for each channel of the colorspace were se
    %  auto-generated code from the colorThresholder App. The colorspace
    %  App and result in a binary mask BW and a composite image maskedRG
    %  which shows the original RGB image values under the mask BW.

    [BW,maskedRGBImage]=createMask(rgbImage, 70, 255);
    %     figure, imshow(BW);
    %     BB= imfill(BW,'holes');
    %     figure,imshow(BB);
    BW_Sobel= edge(BW,'Sobel');
    %     figure, imshow(BW_Sobel);
    %     BB_Sobel= edge(BB,'Sobel');
```

nd Window









```
%% Velocity calculation

fig = gcf; % current figure handle
%'Choose a set of points. A shift-, right-, or double-click adds a final point and ends the selection';
[x,y] = getpts(fig);
warndlg('Choose a set of points. A shift-, right-, or double-click adds a final point and ends the select:

% [x, y] = getpts(fig) lets you choose a set of points in the current axes of figure fig using the mouse.
% Coordinates of the selected points are returned in X and Y.
% Use normal button clicks to add points.
% A shift-, right-, or double-click adds a final point and ends the selection.
% Pressing Return or Enter ends the selection without adding a final point. Pressing Backspace or Delete r

x_cm=x*dim_pixel;
y_cm=y*dim_pixel;
delta_t=time_gap/fps; % 25 fps
for i = 2:length(x)
    Vel(i)=sqrt((x_cm(i)-x_cm(i-1))^2+(y_cm
end
Vel'
Vel_media=mean(Vel')

%% SAVE ALL
save data_edgelist_1008_1
```