REFERENCES

Adrian, R. J. (1991). Particle-imaging techniques for experimental fluid mechanics. *Annual Reviews of Fluid Mechanics,* **23**, 261-304.

Aryafar, H. and H. P. Kavehpour (2006). Drop coalescence through planar surfaces. *Physics of Fluids,* **18**, 072015.

Bartolo, D., C. Josserand and D. Bonn (2006). Singular jets and bubbles in drop impact. *Physical Review Letters,* **96**, 124501-124504.

Batchelor, G. K. (1967). *An introduction to fluid dynamics.* London: Cambridge University Press

Birkhoff, G., D. MacDougall, E. Pugh and G. I. Taylor (1948). Explosives with lined cavities. *Journal of Applied Physics,* **19**, 563-582.

Birkhoff, G. (1956). Stability of spherical bubbles. *Quarterly of Applied Mathematics,* **13(4)**, 451-453.

Blanchard, D. C. and A. H. Woodcock (1957). Bubble formation in the sea and its meteorological significance. *Tellus,* **9**, 150-158.

Blanchard, D. C. and L. D. Syzdek (1981). Bubble scavenging of bacteria in freshwater quickly produces bacterial enrichment in airborne jet drops. *Limnol Oceanography,* **26(5)**, 961-964.

Blanchard, D. C. and L. D. Syzdek (1988). Film drop production as a function of bubble size. *Journal of geophysical research,* **93(C4)**, 3649-3654.

Blanchard, D. C. (1989). The size and height to which jet drops are ejected from bursting bubbles in seawater. *Journal of geophysical research,* **94(C8)**, 10,999-911,002.

Blanchard, D. C. (1990). Surface-active monolayers, bubbles and jet drops. *Tellus,* **42B**, 200-205.

Cai, Y. K. (1989). Phenomena of a liquid drop falling to a liquid surface. *Experiments in Fluids,* **7**, 388-394.

Carroll, K. and R. Mesler (1981). Splashing liquid drops form vortex rings and not jets at low froude numbers. *Journal of Applied Physics,* **52(1)**, 507.

Chapman, D. S. and P. R. Critchlow (1967). Formation of vortex rings from falling drops. *Journal of Fluid Mechanics,* **29(1)**, 177-185.

Charles, G. E. and S. G. Mason (1960). The mechanism of partial coalescence of liquid drops at liquid/liquid interfaces. *Journal of Colloid Science,* **15**, 105-122.

Cole, D. E. (2007). Unpublished data.

Cossali, G. E., M. Marengo, A. Coghe and S. Zhdanov (2004). The role of time in single drop splash on thin film. *Experiments in Fluids,* **36(6)**, 888-900.

Cresswell, R. W. and B. R. Morton (1995). Drop-formed vortex rings - the generation of vorticity. *Physics of Fluids,* **7(6)**, 1363-1370.

Dantec Dynamics (2000). *Flowmanager software and introduction to piv instrumentation user manual.* Tonsbakken: Dantec Dynamics

Davidson, M. R. (1998). Prediction of crown formation following the impact of a drop on a thin liquid film. *13th Australasian Fluid Mechanics Conference. Monash University, Melbourne, Australia*

Davis Philip, A. G. (1960). Effect of viscosity on the splash caused by a drop of water falling into a mixture of glycerol and water. *Journal of Applied Physics,* **31(4)**, 727-730.

Durst, F. (1996). Penetration length and diameter of vortex rings generated by impacting water drops. *Experiments in Fluids,* **21**, 110-117.

Edgerton, H. E. and J. R. Killian (1979 Reprint). *Moments of vision: Stroboscopic revolution in photography.* Cambridge, Massachaussets: MIT Press

Ellison, W. D. (1944). Studies of raindrop erosion. *Agricultural Engineering,* **25**, 131-136.

Elmore, P. A., G. L. Chahine and H. N. Oguz (2001). Cavity and flow measurements of reproducible bubble entrainment following drop impacts. *Experiments in Fluids,* **31(6)**, 664-673.

Engel, O. G. (1955). Waterdrop collisions with solid surfaces. *Journal of Research of the National Bureau of Standards,* **54(5)**, 281-298.

Engel, O. G. (1966). Crater depth in fluid impacts. *Journal of Applied Physics,* **37(4)**, 1798-1808.

Engel, O. G. (1967). Initial pressure, initial flow velocity and the time dependence of crater depth in fluid impacts. *Journal of Applied Physics,* **38(10)**, 3935-3940.

Esmailizadeh, L. and R. Mesler (1985). Bubble entrainment with drops. *Journal of colloid and interface science,* **110(2)**, 561-574.

Fedorchenko, A. I. and A. B. Wang (2004). On some common features of drop impact on liquid surfaces. *Physics of Fluids,* **16(5)**, 1349-1365.

Ferreira, A. G. and M. J. Singer (1985). Energy dissipation for water drop impact into shallow pools. *Soil Science Society Of America Journal,* **49**, 1537-1542.

Fitt, B. and M. Shaw (1989). Transports of blight. *New Scientist,* **12**, 25-27.

Franz, G. J. (1959). Splashes as sources of sound in liquids. *Journal of the Acoustical Society of America,* **31(8)**, 1080-1096.

Hallett, J. and L. Christensen (1984). Splash and penetration of drops in water. *Journal de Recherches Atmospheriques,* **18(4)**, 225-242.

Hobbs, P. V. and T. Osheroff (1967). Splashing of drops on shallow liquids. *Science,* **158**, 1184-1186.

Honey, E. M. and H. P. Kavehpour (2006). Astonishing life of a coalescing drop on a free surface. *Physical Review E,* **73(2)**,

Hsiao, M., S. Lichter and L. G. Quintero (1988). The critical weber number for vortex and jet formation for drops impinging on a liquid pool. *Physics of Fluids,* **31(12)**, 3560-3562.

Jayaratne, O. W. and B. J. Mason (1964). The coalescence and bouncing of water drops at an air/water interface. *Proceedings of the Royal Society London,* **280A**, 545-565.

Leneweit, G., R. Koehler, K. G. Roesner and G. Schafer (2005). Regimes of drop morphology in oblique impact on deep fluids. *Journal of Fluid Mechanics,* **543**, 303-331.

Levin, Z. and P. V. Hobbs (1971). Splashing of water drops on solid and wetted surfaces hydrodynamics and charge separation. *Philosophical Transactions of the Royal Society London A,* **269**, 555-585.

Liow, J. L., W. H. R. Dickinson, M. J. Allan and N. B. Gray (1994). Study of slopping and splashing in a cylindrical bath with top-submerged injection. *Metallurgical and Materials Transactions B,* **26B**, 887-889.

Liow, J. L. (2001). Splash formation by spherical drops. *Journal of Fluid Mechanics,* **427**, 73-105.

Machu, G., W. Meile, L. C. Nitsche and U. Schaflinger (2001). Coalescence, torus formation and breakup of sedimenting drops: Experiments and computer simulations. *Journal of Fluid Mechanics,* **447**, 299-336.

Macklin, W. C. and P. V. Hobbs (1969). Subsurface phenomena and the splashing of drops on shallow liquids. *Science,* **166**, 107-108.

Macklin, W. C. and G. J. Mextaxas (1976). Splashing of drops on liquid layers. *Journal of Applied Physics,* **47(9)**, 3963-3970.

Manzello, S. L. and J. C. Yang (2003). The influence of liquid pool temperature on the critical impact weber number for splashing. *Physics of Fluids,* **15(1)**, 257-260.

Mehdi-Nejad, V., J. Mostaghimi and S. Chandra (2003). Air bubble entrapment under a impacting droplet. *Physics of Fluids,* **15(1)**, 173-183.

Melling, A. (1997). Tracer particles and seeding for particle image velocimetry. *Measurement Science and Technology,* **8(12)**, 1406.

Meynart, R. (1983). Speckle velocimetry study of vortex pairing in low-re unexcited jet. *Physics of Fluids,* **26(8)**, 2074-2079.

Mohamed-Kassim, Z. and E. K. Longmire (2003). Drop impact on a liquid-liquid interface. *Physics of Fluids,* **15(11)**, 3263-3273.

Morton, B. R. and R. W. Cresswell (1992). Raindrops in the sea i - generation of vorticity and vortex ring production. *11th Australasian Fluid Mechanics Conference. University Of Tasmania, Hobart, Australia*

Morton, D., F. Augier, O. Masbernat and M. Rudman (2000). A numerical and experimental study of vortex ring formation resulting from impacting drops. *FEDSM. Boston, Massachusetts*

Morton, D., M. Rudman and J. L. Liow (2000). An investigation of the flow regimes resulting from splashing drops. *Physics of Fluids,* **12(4)**, 747-763.

Munson, B. R., F. D. Young and T. H. Okiishi (1998). *Fundamentals of fluid mechanics.* John Wiley

Mutchler, C. K. and L. M. Hansen (1970). Splash of a waterdrop at terminal velocity. *Science,* **169**, 1311-1312.

Neitzel, G. P. and P. Dell'Aversana (2002). Noncalescence and nonwetting behavior of liquids. *Annual Review of Fluid Mechanics,* **34**, 267-289.

Oguz, H. N. and A. Prosperetti (1989). Surface-tension effects in the contact of liquid surfaces. *Journal of Fluid Mechanics,* **203**, 149-171.

Oguz, H. N. and A. Prosperetti (1990). Bubble entrainment by the impact of drops on liquid surfaces. *Journal of Fluid Mechanics,* **219**, 143-179.

Oguz, H. N. and A. Prosperetti (1991). Numerical calculation of the underwater noise of rain. *Journal of Fluid Mechanics,* **228**, 417-442.

Pandit, A. B. and J. F. Davidson (1990). Hydrodynamics of the rupture of thin liquid films. *Journal of Fluid Mechanics,* **212**, 11-24.

Peck, B. and L. Sigurdson (1994). The three-dimensional vortex structure of an impacting water drop. *Physics of Fluids,* **6(2)**, 564-576.

Peck, B. and L. Sigurdson (1995). The vortex ring velocity resulting from an impacting water drop. *Experiments in Fluids,* **18**, 351-357.

Pikhitsa, P. and A. Tsargorodskaya (2000). Possible mechanism for multistage coalesces of a floating droplet on the air/liquid interface. *Colloids and Surfaces A: Physicochemical and Engineering Aspects,* **167**, 287-291.

Prasad, A. K. (2000). Particle image velocimetry. *Current Science,* **79(1)**, 51-60.

Prosperetti, A. and H. N. Oguz (1993). The impact of drops on liquid surfaces and the underwater noise of rain. *Annual Review of Fluid Mechanics,* **25**, 577-602.

Pumphrey, H. C. and L. A. Crum (1989). Underwater sound produced by individual drop impacts and rainfall. *Journal of the Acoustical Society of America,* **85(4)**, 1518-1526.

Pumphrey, H. C. and P. A. Elmore (1990). The entrainment of bubbles by drop impacts. *Journal of Fluid Mechanics,* **220**, 539-567.

Raffel, M., C. Willert and J. Kompenhans (1998). *Particle image velocimetery: A practical guide.* Springer-Verlag

Rein, M. (1993). Phenomena of liquid-drop impact on solid and liquid surfaces. *Fluid Dynamics Research,* **12(2)**, 61-93.

Rein, M. (1996). The transitional regime between coalescing and splashing drops. *Journal of Fluid Mechanics,* **306**, 145-165.

Reynolds, O. (1875). On the action of rain to calm the sea. *Memoirs & Proceedings of the Manchester Literary & Philosophical Society,* **14**, 72-74.

Reynolds, O. (1881). On drops floating on the surface of water. *Proceedings of the Manchester Literary and Philosophical Society,* **21**, 1-2.

Rieber, M. and A. Frohn (1999). A numerical study on the mechanism of splashing. *International journal of heat and fluid flow,* **20**, 455-461.

Rioboo, R., C. Bauthier, J. Conti, M. Voue and J. De Coninck (2003). Experimental investigation of splash and crown formation during single drop impact on wetted surfaces. *Experiments in Fluids,* **35(6)**, 648-652.

Rodriguez, F. and R. Mesler (1987). The penetration of drop formed vortex rings into pools of liquid. *Journal of colloid and interface science,* **121(1)**, 121-129.

Roisman, I. V. and C. Tropea (2002). Impact of a drop onto a wetted wall: Description of crown formation and propagation. *Journal of Fluid Mechanics,* **472**, 373-397.

Scrimger, J. A. (1985). Underwater noise caused by precipitation. *Nature,* **318**, 647-649.

Scrimger, J. A., D. J. Evans, G. A. McBean, D. M. Farmer and R. B. Kerman (1987). Underwater noise due to rain, hail and snow. *Journal of the Acoustical Society of America,* **81(1)**, 79-86.

Shinohara, K., Y. Sugii, A. Aota, A. Hibara, M. Tokeshi, T. Kitamori and K. Okamoto (2004). High-speed micro-piv measurements of transient flow in microfluidic devices. *Measurement Science and Technology,* **15(10)**, 1965.

Sigler, J. and R. Mesler (1990). The behaviour of the gas film formed upon drop impact with a liquid surface. *Journal of colloid and interface science,* **134(2)**, 459-474.

Sivakumar, D. and C. Tropea (2002). Splashing impact of a spray onto a liquid film. *Physics of Fluids,* **14(12)**, L85-L88.

Spiel, D. E. (1994). The number and size of jet drops produced by air bubbles bursting on a fresh water surface. *Journal of geophysical research,* **99(C5)**, 10,289-210,296.

Spiel, D. E. (1995). On the births of jet drops from bubbles bursting on water surfaces. *Journal of geophysical research,* **100(C3)**, 4995-5006.

Spiel, D. E. (1997). More on the births of jet drops from bubbles bursting on seawater surfaces. *Journal of geophysical research,* **102(C3)**, 5815-5821.

Thomson, J. J. (1885). On the formation of vortex rings by drops falling into liquids and some allied phenomena. *Proceedings of the Royal Society London*, 417-436.

Thoroddsen, S. T. and K. Takehara (2000). The coalescence cascade of a drop. *Physics of Fluids,* **12(6)**, 1265-1267.

Thoroddsen, S. T. (2002). The ejecta sheet generated by the impact of a drop. *Journal of Fluid Mechanics,* **451**, 373-381.

Thoroddsen, S. T., T. G. Etoh and K. Takehara (2003). Air entrapment under an impacting drop. *Journal of Fluid Mechanics,* **478**, 125-134.

Towers, D. P. and C. E. Towers (2004). Cyclic variability measurements of in-cylinder engine flows using high-speed particle image velocimetry. *Measurement Science and Technology,* **15(9)**, 1917-1925.

Tuong, T. P. and D. J. Painter (1974). The interaction between a raindrop and a shallow body of water. *5th Australiasian Conference on Hydraulics and Fluid Mechanics. University of Canterbury, Christchurch, New Zealand*

Vander-Wal, R. L., G. M. Berger and S. D. Mozes (2006). Droplets splashing upon films of the same fluid of various depths. *Experiments in Fluids,* **40**, 33-52.

Wang, A. B. and C. C. Chen (2000). Splashing impact of a single drop onto very thin liquid films. *Physics of Fluids,* **12(9)**, 2155-2158.

Weiss, D. A. and A. L. Yarin (1999). Single drop impact onto liquid films: Neck distortion, jetting, tiny bubble entrainment, and crown formation. *Journal of Fluid Mechanics,* **385**, 229-254.

Worthington, A. M. (1897). Impact with a liquid surface, studied by the aid of instantaneous photography: I. *Philosophical Transactions of the Royal Society London,* **189A**, 137-148.

Worthington, A. M. (1900). Impact with a liquid surface studied by the aid of instantaneous photography: Ii. *Philosophical Transactions of the Royal Society London,* **194A**, 175-199.

Worthington, A. M. (1908). *A study of splashes.* Longmans, Green, and Co.

Yarin, A. L. and D. A. Weiss (1995). Impact of drops on solid surfaces: Self-similar capillary waves, and splashing as a new type of kinematic discontinuity. *Journal of Fluid Mechanics,* **283**, 141-173.

APPENDIX A – MATLAB CODE

```
function drop_gui(input_str)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRAM DESCRIPTION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% This purpose of this matlab program is to easily and quickly calculate
% the diameter and velocity of falling drops. The drops are extracted from
% a source image by applying a threshold value to a grayscale image to
% convert it into a series of black or white pixels. The image is then
% cleaned up and analysed. The user gets IMMEDIATE feedback on how
% different parameters effect the final result.
%
% CURRENT PROBLEMS
% - Lots of error catching bug fixes required
% - Program flow is a bit all over the place, needs to be tidied up by
% combinding functions into 1 instead of having 2 seperate ones
% - Have to use global variables for everything otherwise they get deleted.
% Not sure there is anyway around this
%
% IMPROVEMENTS REQUIRED
% - Write out the results to file, so they can be easily viewed in excel
% - Implent other more accurate edge detection methods
% - Implement the autothreshold feature
% - Implement a more accurate method of diameter determination. Use the
% profile of the drop to revolve it and get a volume out, then relate it
% back to a diameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all

% Define some global variables so information is not destroyed when
% the function finishes. Need to hold all the gui handles cause the findobj
% function in matlab is just way too slow

% Variables to hold the image information
global drop1_org_image drop1_threshold_image drop1_clean_image
global drop2_org_image drop2_threshold_image drop2_clean_image

% Handles for all the image boxes
global h_drop1_org h_drop1_threshold h_drop1_clean h_drop1_final
global h_drop2_org h_drop2_threshold h_drop2_clean h_drop2_final

% Handles for the text boxes etc in the drop panels
global h_drop1_invert_check h_drop1_threshold_slider h_drop1_threshold_slider_textbox
h_drop1_dropdown
global h_drop2_invert_check h_drop2_threshold_slider h_drop2_threshold_slider_textbox
h_drop2_dropdown

% Handles for the objects used in the calculation panel
global  h_calc_diameter_textbox h_calc_calibration_editbox h_drop1_remove_size_editbox
global stats1 stats2

% Some variables to handle file names
global drop1_file_name path previous_path path
```

```
global boundary
global stats1 stats2

% Set the way images will be views
iptsetpref('ImshowBorder','tight');
iptsetpref('ImshowAxesVisible','off');
iptsetpref('ImviewInitialMagnification','fit');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% CREATE THE GUI
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% The portion of the program creates all the figures, text boxes etc. etc
% used during the program. The GUI is structured as follows Main Figure -->
% 3 uipanel objects --> Buttons, Axes and Boxes children of the UiPanel
% objects. This allows the positioning of everything to be set relative to
% the panels not the main figure.
% ALL SPACING IS IN NORMALISED OR PERCENT TERMS
%
% CURRENT PROBLEMS
% - Stupid MATLAB won't allow me to have the axes as children of the
% uipanel obect when imshow is executed. Bug in MATLAB. Only work around is
% to set the positions of the Axes relative to the main figure
%
% IMPROVEMENTS REQUIRED
% - Organise things a bit better, i.e. the naming of things need to follow
% a better convention then what I use now
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%

% Extract screen size and dimensions required for the figure
figure_width = 1024;
figure_height = 768;
screen_size = get(0,'ScreenSize');

% Create main figure and add all the gui elements to it. Also centre
% the GUI
h_fig  =  figure('position',[screen_size(3)/2-figure_width/2  screen_size(4)/2-figure_height/2  figure_width
figure_height ],...
                    'menubar', 'figure',...
                    'color', 'white',...
                    'name','Drop Velocity and Diameter Calculator',...
                    'resize','off',...
                    'numbertitle','off');


% Split the screen up into 3 panels for clarity. First two panels
% display all the image and the third one does the calculations

% Define a spacing height in percent
panel_height = 0.4;
h_drop1_panel = uipanel('position', [0 1-panel_height 1 panel_height],...
                    'title','Drop 1 Analysis Panel',...
                    'parent',h_fig);

h_drop2_panel = uipanel('position', [0 1-2*panel_height 1 panel_height],...
                    'title','Drop 2 Analysis Panel');
```

```
h_calc_panel = uipanel('position', [0 0 1 1-2*panel_height],...
                        'title','Calcuation Panel');


% Need to figure out where the image boxes and labels will lie relative to
% the border. Als need to find how big the boxes will be if the resolution
% is changed
side_buffer = 0.1;
bottom_buffer = 0.2;
top_buffer = 0.1;

image_box_spacing = 0.01;
image_box_width = (1 - 2*side_buffer - 3*image_box_spacing)/4;
image_box_height = (1 - top_buffer - bottom_buffer);
left_centre_offset = side_buffer+image_box_width/2;

% Create all the axes objects now
% DROP1 CALCULATION BOX
h_drop1_org = axes('units','normalized',...
                        'position', [0.01 0.67 0.25 0.3],...
                        'parent',h_fig,...
                        'visible','off',...
                        'box','on',...
                        'activepositionproperty','position');
                        %'PlotBoxAspectRatio',[1 1 1],...
                        %'DataAspectRatio',[1 1 1],...
                    %                       'DataAspectRatio',[1 1 1],...
 %                        'DataAspectRatioMode','manual');

h_drop1_threshold = axes('units','normalized',...
                        'position', [0.26 0.67 0.23 0.3],...
                        'parent',h_fig,...
                        'visible','off',...
                        'box','on',...
                        'PlotBoxAspectRatio',[1 1 1]);

h_drop1_clean = axes('units','normalized',...
                        'position', [0.51 0.67 0.23 0.3],...
                        'parent',h_fig,...
                        'visible','off',...
                        'box','on',...
                        'PlotBoxAspectRatio',[1 1 1]);

h_drop1_final = axes('units','normalized',...
                        'position', [0.76 0.67 0.23 0.3],...
                        'parent',h_fig,...
                        'visible','off',...
                        'box','on',...
                        'PlotBoxAspectRatio',[1 1 1]);

% DROP2 CALCULATION BOX
h_drop2_org = axes('units','normalized',...
                        'position', [0.01 0.27 0.23 0.3],...
                        'parent',h_fig,...
                        'visible','off',...
                        'box','off',...
                        'activepositionproperty','position',...
                        'PlotBoxAspectRatio',[1 1 1]);
```

```
h_drop2_threshold = axes('units','normalized',...
                        'position', [0.26 0.27 0.23 0.3],...
                        'parent',h_fig,...
                        'visible','off',...
                        'box','on',...
                        'PlotBoxAspectRatio',[1 1 1]);

h_drop2_clean = axes('units','normalized',...
                        'position', [0.51 0.27 0.23 0.3],...
                        'parent',h_fig,...
                        'visible','off',...
                        'box','on',...
                        'PlotBoxAspectRatio',[1 1 1]);

h_drop2_final = axes('units','normalized',...
                        'position', [0.76 0.27 0.23 0.3],...
                        'parent',h_fig,...
                        'visible','off',...
                        'box','on',...
                        'PlotBoxAspectRatio',[1 1 1]);

%% Create all the buttons and text boxes. ALL THESE CONTROLS ARE INTERACTIVE
% DROP1 PANEL ITEMS
h_drop1_open_button = uicontrol('units','normalized',...
                        'style','pushbutton',...
                        'string','Open',...
                        'parent',h_drop1_panel,...
                        'position',[0.075 0.05 0.1 0.1],...
                        'callback',@open_image1);

h_drop1_invert_check = uicontrol('units','normalized',...
                        'style','checkbox',...
                        'value',0,...
                        'string','Invert',...
                        'position',[0.275 0.05 0.1 0.05],...
                        'parent',h_drop1_panel,...
                        'callback',@threshold_image1);

h_drop1_threshold_slider = uicontrol('units','normalized',...
                        'style','slider',...
                        'position',[0.325 0.08 0.15 0.05],...
                        'max',1,...
                        'min',0,...
                        'sliderstep',[0.025 0.05],...
                        'value',0.4,...
                        'callback',@threshold_image1,...
                        'parent',h_drop1_panel);

h_drop1_threshold_2_slider = uicontrol('units','normalized',...
                        'style','slider',...
                        'position',[0.325 0.03 0.15 0.05],...
                        'max',1,...
                        'min',0,...
                        'sliderstep',[0.025 0.05],...
                        'value',0.3,...
                        'callback',@threshold_image1,...
                        'parent',h_drop1_panel);

h_drop1_threshold_slider_textbox = uicontrol('units','normalized',...
                        'style','text',...
```

```
                    'position',[0.375 0.13 0.03 0.05],...
                    'string',get(h_drop1_threshold_slider,'value'),...
                    'parent',h_drop1_panel);

h_drop1_threshold_2_slider_textbox = uicontrol('units','normalized',...
                    'style','text',...
                    'position',[0.4 0.13 0.05 0.05],...
                    'string',get(h_drop1_threshold_slider,'value'),...
                    'parent',h_drop1_panel);

h_drop1_remove_size_editbox = uicontrol('units','normalized',...
                    'style','edit',...
                    'position',[0.65 0.05 0.075 0.075],...
                    'string','10',...
                    'backgroundcolor',[1 1 1],...
                    'parent',h_drop1_panel,...
                    'callback',@clean_image1);

h_drop1_dropdown = uicontrol('units','normalized',...
                    'style','popupmenu',...
                    'position',[0.9 0.05 0.075 0.075],...
                    'string','2',...
                    'backgroundcolor',[1 1 1],...
                    'parent',h_drop1_panel,...
                    'callback',@calculate_data);

% DROP2 PANEL ITEMS
h_drop2_open_button = uicontrol('units','normalized',...
                    'style','pushbutton',...
                    'string','2nd Image',...
                    'parent',h_drop2_panel,...
                    'position',[0.075 0.05 0.1 0.1],...
                    'callback',@open_image2,...
                    'enable','off');

h_drop2_invert_check = uicontrol('units','normalized',...
                    'style','checkbox',...
                    'value',0,...
                    'string','Invert',...
                    'position',[0.275 0.05 0.1 0.05],...
                    'parent',h_drop2_panel,...
                    'callback',@threshold_image2);

h_drop2_threshold_slider = uicontrol('units','normalized',...
                    'style','slider',...
                    'position',[0.325 0.08 0.15 0.05],...
                    'max',1,...
                    'min',0,...
                    'sliderstep',[0.025 0.05],...
                    'value',0.4,...
                    'callback',@threshold_image2,...
                    'parent',h_drop2_panel);

h_drop2_threshold_2_slider = uicontrol('units','normalized',...
                    'style','slider',...
                    'position',[0.325 0.03 0.15 0.05],...
                    'max',1,...
                    'min',0,...
                    'sliderstep',[0.025 0.05],...
                    'value',0.3,...
```

```
                    'callback',@threshold_image2,...
                    'parent',h_drop2_panel);

h_drop2_threshold_slider_textbox = uicontrol('units','normalized',...
                    'style','text',...
                    'position',[0.375 0.13 0.03 0.05],...
                    'string',get(h_drop2_threshold_slider,'value'),...
                    'parent',h_drop2_panel);

h_drop2_threshold_2_slider_textbox = uicontrol('units','normalized',...
                    'style','text',...
                    'position',[0.4 0.13 0.05 0.05],...
                    'string',get(h_drop2_threshold_slider,'value'),...
                    'parent',h_drop2_panel);

h_drop2_remove_size_editbox = uicontrol('units','normalized',...
                    'style','edit',...
                    'position',[0.65 0.05 0.075 0.075],...
                    'string','10',...
                    'backgroundcolor',[1 1 1],...
                    'parent',h_drop2_panel,...
                    'callback',@clean_image2);

h_drop2_dropdown = uicontrol('units','normalized',...
                    'style','popupmenu',...
                    'position',[0.9 0.05 0.075 0.075],...
                    'string','2',...
                    'backgroundcolor',[1 1 1],...
                    'parent',h_drop2_panel,...
                    'callback',@calculate_data);

% CALCULATION PANEL ITEMS
h_calc_velocity_textbox = uicontrol('units','normalized',...
                    'style','text',...
                    'position',[0.815 0.1 0.075 0.1],...
                    'string','',...
                    'parent',h_calc_panel,...
                    'backgroundcolor',[1 1 1]);

h_calc_diameter1_textbox = uicontrol('units','normalized',...
                    'style','text',...
                    'position',[0.815 0.7 0.075 0.1],...
                    'string','',...
                    'parent',h_calc_panel,...
                    'backgroundcolor',[1 1 1]);

h_calc_diameter2_textbox = uicontrol('units','normalized',...
                    'style','text',...
                    'position',[0.815 0.5 0.075 0.1],...
                    'string','',...
                    'parent',h_calc_panel,...
                    'backgroundcolor',[1 1 1]);

h_calc_average_diameter_textbox = uicontrol('units','normalized',...
                    'style','text',...
                    'position',[0.815 0.3 0.075 0.1],...
                    'string','',...
                    'parent',h_calc_panel,...
                    'backgroundcolor',[1 1 1]);
```

```
h_calc_eccentricity1_textbox = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.9 0.7 0.075 0.1],...
                        'string','',...
                        'parent',h_calc_panel,...
                        'backgroundcolor',[1 1 1]);

h_calc_eccentricity2_textbox = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.9 0.5 0.075 0.1],...
                        'string','',...
                        'parent',h_calc_panel,...
                        'backgroundcolor',[1 1 1]);

h_calc_average_eccentricity_textbox = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.9 0.3 0.075 0.1],...
                        'string','',...
                        'parent',h_calc_panel,...
                        'backgroundcolor',[1 1 1]);

h_calc_framerate_editbox = uicontrol('units','normalized',...
                        'style','edit',...
                        'position',[0.5 0.5 0.1 0.1],...
                        'string',num2str(2000),...
                        'parent',h_calc_panel,...
                        'backgroundcolor',[1 1 1],...
                        'callback',@calculate_data);

h_calc_sigma_editbox = uicontrol('units','normalized',...
                        'style','edit',...
                        'position',[0.5 0.3 0.1 0.1],...
                        'string','1',...
                        'parent',h_calc_panel,...
                        'backgroundcolor',[1 1 1],...
                        'callback',@threshold_image1);

h_calc_calibration_editbox = uicontrol('units','normalized',...
                        'style','edit',...
                        'position',[0.35 0.5 0.15 0.1],...
                        'string',num2str(0.022),...
                        'parent',h_calc_panel,...
                        'backgroundcolor',[1 1 1],...
                        'callback',@calculate_data);

h_calc_location_editbox = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.01 0.8 0.3 0.1],...
                        'string','',...
                        'parent',h_calc_panel,...
                        'backgroundcolor',[1 1 1]);

h_calc_analysis_type = uicontrol('units','normalized',...
                        'style','popupmenu',...
                        'position',[0.01 0.6 0.1 0.1],...
                        'string','Canny|Threshold|Sobel|Prewitt|Laplacian|Zero-Cross',...
                        'parent',h_calc_panel);


%% Create all the NON-INTERACTIVE text boxes
```

```
h_drop1_remove_size_label = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.5 0.03 0.15 0.1],...
                        'string','Size in pixels of the smallest object to be removed',...
                        'parent',h_drop1_panel);

h_drop1_regionpick_label = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.75 0.03 0.15 0.1],...
                        'string','Which regions properties do you wish to analyse?',...
                        'parent',h_drop1_panel);

h_drop2_remove_size_label = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.5 0.03 0.15 0.1],...
                        'string','Size in pixels of the smallest object to be removed',...
                        'parent',h_drop2_panel);

h_drop2_regionpick_label = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.75 0.03 0.15 0.1],...
                        'string','Which regions properties do you wish to analyse?',...
                        'parent',h_drop2_panel);


h_calc_velocity_label = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.7 0.1 0.1 0.1],...
                        'string','Velocity (m/s)',...
                        'parent',h_calc_panel);

h_calc_diameter_label = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.8 0.8 0.1 0.1],...
                        'string','Diameter (mm)',...
                        'parent',h_calc_panel);

h_calc_eccentricty_label = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.9 0.8 0.075 0.1],...
                        'string','Eccentricity',...
                        'parent',h_calc_panel);

h_calc_drop1_label = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.7 0.7 0.1 0.1],...
                        'string','Drop 1',...
                        'parent',h_calc_panel);

h_calc_drop2_label = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.7 0.5 0.1 0.1],...
                        'string','Drop 2',...
                        'parent',h_calc_panel);

h_calc_average_label = uicontrol('units','normalized',...
                        'style','text',...
                        'position',[0.7 0.3 0.1 0.1],...
                        'string','Average',...
```

```
                    'parent',h_calc_panel);

h_calc_framerate_label = uicontrol('units','normalized',...
                    'style','text',...
                    'position',[0.5 0.6 0.1 0.1],...
                    'string','Frame Rate (FPS)',...
                    'parent',h_calc_panel);

h_calc_calibration_label = uicontrol('units','normalized',...
                    'style','text',...
                    'position',[0.35 0.6 0.15 0.1],...
                    'string','Calibration Factor (mm/pixel)',...
                    'parent',h_calc_panel);
%obj = 0;
%event = 0;
%open_image1(obj,event)

previous_path = '*.tif';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% OPEN THE INPUT IMAGE/IMAGES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% This function opens up the image after the user has clicked the open
% button. A UI for getting the file is displayed.
%
%
% CURRENT PROBLEMS
% - If other objects are clicked before you open an image silly errors are
% generated
%
% IMPROVEMENTS REQUIRED
% - Be good if the last location opened could be stored in a text file so
% it can be recalled at a later date
% - Need to add a button that allows you to pick how many frames ahead you
% want to skip
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%

function open_image1(obj,event)


  [drop1_file_name, path] = uigetfile(previous_path,'Open Image 1');
  drop1_full_name = strcat(path, drop1_file_name);
  drop1_org_image = imread(drop1_full_name);

  %structure_ele = strel('disk',2);
  %drop1_org_image = imtophat(drop1_org_image,structure_ele);
  %drop1_org_image = imadjust(drop1_org_image,[0 0.01],[0 0]);

  previous_path = strcat(path,'*.tif');
  set(h_calc_location_editbox,'string',strcat(path,drop1_file_name));

  % Find the figure to put the original image in and show it
  set(gcf,'CurrentAxes',h_drop1_org);
  imshow(drop1_org_image);

  % Start the threshold function which will create the binary image
  threshold_image1(obj,event);
```

```
    % Need to find what the next image file will be called so we can call
    % it straight up
    filename_length = length(drop1_file_name);

    %Find where the first digit of the incrementing number starts
    start_number = filename_length - 4 - 5;
    new_frame_number = str2num(drop1_file_name(start_number:start_number+5)) + 1;

    % Convert the number into a string and add leading zeros to the start
    new_filename = num2str(new_frame_number);

        while length(new_filename) ~= 6

            new_filename = strcat('0',new_filename);

        end

    % Put it all together all read in the next frame
    final_filename = strcat(drop1_file_name(1:start_number-1),new_filename,'.tif');
    drop2_full_name = strcat(path, final_filename);
    drop2_org_image = imread(drop2_full_name);

    % Find the figure to put the original image in and show it
    set(gcf,'CurrentAxes',h_drop2_org);
    imshow(drop2_org_image);

    % Start the threshold function which will create the binary image
    threshold_image2(obj,event);



end

%function open_image2(obj,event)

 %   drop2_file = uigetfile('*.tif','Open Image 2')
  %  drop2_org_image = imread(drop2_file);

    % Find the figure to put the original image in and show it
 %   set(gcf,'CurrentAxes',h_drop2_org);
  %  imshow(drop2_org_image);

    % Start the threshold function which will create the binary image
   % threshold_image2(obj,event);

%end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% THRESHOLD THE IMAGE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% This function essentially takes the raw image and turns every pixel into
% either a fully black or fully white pixel. The cut off limit for the
% transition between black and white is controlled by THRESHOLD_VALUE when
% the user inputs via the slider
%
%
% CURRENT PROBLEMS
```

```
% -
%
% IMPROVEMENTS REQUIRED
% - Allow the changing of the threshold hold so that different methods can
% be used i.e. sobel, canning etc. etc. This would probably require the
% algorthim to be re-written
% - Allow the user to enter in a threshold value into a text box
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
function threshold_image1(obj, event)

    % Get the threshold value from the slider. Also output the value from
    % the slider into the text box above it
    threshold_value = get(h_drop1_threshold_slider,'value');
    threshold_2_value = get(h_drop1_threshold_2_slider,'value');
    set(h_drop1_threshold_slider_textbox,'string',num2str(threshold_value));
    set(h_drop1_threshold_2_slider_textbox,'string',num2str(threshold_2_value));
    sigma_value = str2num(get(h_calc_sigma_editbox,'string'));

    analysis_type = get(h_calc_analysis_type,'value');

    % Do the edge detection of the image based on which analysis type is
    % selected
    switch analysis_type
        case 1
            drop1_threshold_image = edge(drop1_org_image,'canny',[threshold_2_value
threshold_value],sigma_value);
        case 2
            drop1_threshold_image = im2bw(drop1_org_image,threshold_value);
        case 3
            drop1_threshold_image = edge(drop1_org_image,'sobel',threshold_value);
        case 4
            drop1_threshold_image = edge(drop1_org_image,'prewitt',threshold_value);
        case 5
            drop1_threshold_image = edge(drop1_org_image,'log',threshold_value);
        case 6
            drop1_threshold_image = edge(drop1_org_image,'zerocross',threshold_value);
    end


    % Now read off the state of the invert check box and invert the image if it
    % is selected
    invert_box_state=get(h_drop1_invert_check,'value');

    if invert_box_state == 1
        drop1_threshold_image=~drop1_threshold_image;
    end

    % Display the thresholded image
    set(gcf,'CurrentAxes',h_drop1_threshold);
    imshow(drop1_threshold_image);

    clean_image1(obj, event);

end

function threshold_image2(obj, event)

    % Get the threshold value from the slider. Also output the value from
    % the slider into the text box above it
```

```
    threshold_value = get(h_drop2_threshold_slider,'value');
    threshold_2_value = get(h_drop2_threshold_2_slider,'value');
    set(h_drop2_threshold_slider_textbox,'string',num2str(threshold_value));
    set(h_drop2_threshold_2_slider_textbox,'string',num2str(threshold_2_value));

    analysis_type = get(h_calc_analysis_type,'value');

    % Do the edge detection of the image based on which analysis type is
    % selected
    switch analysis_type
        case 1
            %drop2_threshold_image = edge(drop2_org_image,'canny',threshold_value);
            drop2_threshold_image = edge(drop2_org_image,'canny',threshold_value);
        case 2
            drop2_threshold_image = im2bw(drop2_org_image,threshold_value);
    end

    % Now read off the state of the invert check box and invert the image if it
    % is selected
    invert_box_state=get(h_drop2_invert_check,'value');

    if invert_box_state == 1
        drop2_threshold_image=~drop2_threshold_image;
    end

    % Display the thresholded image
    set(gcf,'CurrentAxes',h_drop2_threshold);
    imshow(drop2_threshold_image);

    clean_image2(obj, event);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CLEANUP THE IMAGE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function essentially fills in any gaps in the region and gets rid of
% any small objects
%
% CURRENT PROBLEMS
% -
%
% IMPROVEMENTS REQUIRED
% -
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function clean_image1(obj, event)

    % Read the minimum size value from the text box
    min_size = str2num(get(h_drop1_remove_size_editbox,'string'));

    % Remove all object containing fewer than the specified number in pixels
    drop1_clean_image = bwareaopen(drop1_threshold_image,min_size);

    % Fill any holes, so that regionprops can be used to estimate
    % the area enclosed by each of the boundaries
    drop1_clean_image = imfill(drop1_clean_image,4,'holes');
```

```
    % Fill in any gaps in the image
   % se = strel('disk',2);
    %drop1_clean_image = imclose(drop1_clean_image,se);

    % Output the cleaned up image for the user to see
    set(gcf,'CurrentAxes',h_drop1_clean);
    imshow(drop1_clean_image);

    final_image1(obj, event)

end

function clean_image2(obj, event)

    % Read the minimum size value from the text box
    min_size = str2num(get(h_drop2_remove_size_editbox,'string'));

    % Remove all object containing fewer than the specified number in pixels
    drop2_clean_image = bwareaopen(drop2_threshold_image,min_size);

    % Fill any holes, so that regionprops can be used to estimate
    % the area enclosed by each of the boundaries
    drop2_clean_image = imfill(drop2_clean_image,'holes');

    % Fill in any gaps in the image
   % se = strel('disk',2);
    %drop1_clean_image = imclose(drop1_clean_image,se);

    % Output the cleaned up image for the user to see
    set(gcf,'CurrentAxes',h_drop2_clean);
    imshow(drop2_clean_image);

    final_image2(obj, event)

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% DISPLAY THE FINAL IMAGE WITH REGIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% The original image is displayed in box with the outline and number of the
% regions on it.
%
%
% CURRENT PROBLEMS
% - The text from other regions is kept on screen after the regions have
% changed. I think this will require a screen redraw
%
% IMPROVEMENTS REQUIRED
% - Would be nice if you could click on the axes and it puts the image in a
% large figure so you can investigate it a bit better.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
function final_image1(obj, event)


    drop1_popup_string = '';

    % Find boundaries of all objects in image. If one region has points inside
```

```
   % it, remove those points.
   [B,L] = bwboundaries(drop1_clean_image,'noholes');

   % Draw the original image on the axes ready for the region outlines
   set(gcf,'CurrentAxes',h_drop1_final);
   imshow(drop1_org_image);

   % Draw the region outlines on the original image
   hold on
   for k = 1:length(B)
     boundary = B{k};
     plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2)
   end

   % Extract the centroids so they can be displayed
   % Create statistics of the region properties for analysis
   stats1                                                                                      =
regionprops(L,'Area','Centroid','EquivDiameter','Eccentricity','Orientation','MajorAxisLength','MinorAxis
Length');

   % Calculate the diameter of each droplet in the image
   for k = 1:length(B)

   % Find the equivalent diameter from region properties
   diameter = stats1(k).EquivDiameter;
   % Find the centroid of the image for velocity measurement
   centroid = stats1(k).Centroid;
   angle = stats1(k).Orientation;
   major_length = stats1(k).MajorAxisLength;
   minor_length = stats1(k).MinorAxisLength;
   text(centroid(1),centroid(2),num2str(k),'color',[1 1 1]);

   % Populate the drop down box with all the available regions
   drop1_popup_string = strcat(drop1_popup_string, num2str(k), '|');
   set(h_drop1_dropdown,'string',drop1_popup_string);

   end


 %   calculate_data(obj, event)

end

function final_image2(obj, event)


   drop2_popup_string = '';

   % Find boundaries of all objects in image. If one region has points inside
   % it, remove those points.
   [B,L] = bwboundaries(drop2_clean_image,'noholes');

   % Draw the original image on the axes ready for the region outlines
   set(gcf,'CurrentAxes',h_drop2_final);
   imshow(drop2_org_image);

   % Draw the region outlines on the original image
   hold on
   for k = 1:length(B)
     boundary = B{k};
```

```
      plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2)
    end

    % Extract the centroids so they can be displayed
    % Create statistics of the region properties for analysis
    stats2 = regionprops(L,'Area','Centroid','EquivDiameter','Eccentricity');

    % Calculate the diameter of each droplet in the image
    for k = 1:length(B)

    % Find the equivalent diameter from region properties
    diameter = stats2(k).EquivDiameter;
    % Find the centroid of the image for velocity measurement
    centroid = stats2(k).Centroid;
    text(centroid(1),centroid(2),num2str(k),'color',[1 1 1]);

    % Populate the drop down box with all the available regions
    drop2_popup_string = strcat(drop2_popup_string, num2str(k), '|');
    set(h_drop2_dropdown,'string',drop2_popup_string);

    end


    calculate_data(obj, event)

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% CALCULATE VELOCITY AND DIAMETER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% This function does all the important calcualtions. Initially, the frame
% rate from the XML file is extracted and used to calculate the velocity.
% Then the regions selected by the user are used to pull out all the
% diameter and centroid data
%
%
% CURRENT PROBLEMS
% -
%
% IMPROVEMENTS REQUIRED
% - Allow the user to select whether they use the XML file or not
% - Results should be output to a text file for import in another program.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
function calculate_data(obj, event)

    % Extract the framerate from the XML file
    filename_length = length(drop1_file_name);
    xml_file = strcat(path, drop1_file_name(1:filename_length - 4 - 7), '.xml');
    file_id = fopen(xml_file, 'r');
    xml_data = fscanf(file_id, '%c', 1000);

    frame_rate_start = findstr(xml_data, '<FrameRate>') + 11;
    frame_rate_end = findstr(xml_data, '</FrameRate>') - 1;

    % Fill in the frame rate box with the extracted number
    frame_rate = str2num(xml_data(frame_rate_start:frame_rate_end));
    set(h_calc_framerate_editbox,'string',num2str(frame_rate));
```

```
    % Extract the data for the regions based on what is picked in the popup
    % menu
    dropdown1_value = get(h_drop1_dropdown,'value');
    dropdown2_value = get(h_drop2_dropdown,'value');

    % Extract info from calibration and framerate boxes
    calibration_factor = str2num(get(h_calc_calibration_editbox,'string'));
    frame_rate = str2num(get(h_calc_framerate_editbox,'string'));

    % Calculate the diamter of the drop from the first drop and apply the
    % calibration factor
    diameter1 = stats1(dropdown1_value).EquivDiameter * calibration_factor;
    diameter2 = stats2(dropdown2_value).EquivDiameter * calibration_factor;
    average_diameter = (diameter1 + diameter2)/2;
    set(h_calc_diameter1_textbox,'string',num2str(diameter1));
    set(h_calc_diameter2_textbox,'string',num2str(diameter2));
    set(h_calc_average_diameter_textbox,'string',num2str(average_diameter));

    % Find the eccentricity of the drops and display it
    eccentricity1 = stats1(dropdown1_value).Eccentricity;
    eccentricity2 = stats2(dropdown2_value).Eccentricity;
    average_eccentricity = (eccentricity1 + eccentricity2) / 2;
    set(h_calc_eccentricity1_textbox,'string',num2str(eccentricity1));
    set(h_calc_eccentricity2_textbox,'string',num2str(eccentricity2));
    set(h_calc_average_eccentricity_textbox,'string',num2str(average_eccentricity));


   % Figure out the centroid and display the velocity
    centroid1 = stats1(dropdown1_value).Centroid;
    centroid2 = stats2(dropdown2_value).Centroid;

    displacement = ((centroid1(1)-centroid2(1))^2+(centroid1(2)-centroid2(2))^2)^0.5;
    final_velocity = displacement * calibration_factor * frame_rate / 1000;
    set(h_calc_velocity_textbox,'string',num2str(final_velocity));


end


function export_data()



end



end
```