

---

# Create Position Data for Joints of Rat's Paw to Determine Fine Motor Movement

## Table of Contents

Input .....	1
Output .....	1
Opening Function .....	3
Output Function .....	4
Irrelevant functions (with respect to user and GUI output) .....	5
Function executed upon pressing Begin button .....	5
More irrelevant functions (with respect to user and GUI output) .....	10

This function creates a guided user interface (GUI) that instructs the user in placing markers on the various joints of a rat's paw. It proceeds through the 40 frames following the start frame (see GUIcreateManualPoints\_OpeningFcn), according to the interval set by the variable Interval (default setting is 10 frames, so 5 frames are shown in total when including the start frame). The GUI can be edited as needed by opening the .fig file in GUIDE. The format for calling the function is as follows:

```
output = GUIcreateManualPoints(RatData,i,j,StartFrame)
```

where...

## Input

1. RatData: Structure created by createManualPawData.m with fields titled Date Folders -- All the folders within a rat's raw data folder, one for every day of experimentation Video Files -- A structure containing information about all the .avi files in a given date folder Date -- The date for a given date folder, written out in mm/dd/yy format
2. i: Indicates which date folder in RatData contains video to be analyzed (same as i in createManualPawData.m)
3. j: Indicates which video file in date folder will be analyzed (same as j in createManualPawData.m)

## Output

1. AllFramesMarkerLocData: Cell array containing a # of structures equal to the number of frames analyzed (default = 5). Each structure contained in AllFramesMarkerLocData contains 3 substructures corresponding to the three views used to place markers (Left, Center, Right). Each of those substructures contains 5 tables corresponding to the different categories of markers placed (pellet center, paw center, metacarpal phalanges- proximal phalanges joint, proximal phalanges-middle phalanges joint, middle phalanges-distal phalanges joint). Finally, each of these tables in turn contains coordinates, either simply one pair (in the case of pellet center and paw center) or four pairs corresponding to the four digits of the rat's paw (Index, Middle, Ring, Pinky; in that order).

Create Position Data for  
Joints of Rat's Paw to Deter-  
mine Fine Motor Movement

---

Other Outputs to Base Workspace (but not the actual output of function)

1. LastMarkedMarkerData: Temporarily keeps track of the markers completed for a given region of a frame thus far. Resets when a frame region is completed. If function is completed correctly, it should match the data contained in the structure for the right region of the end frame.
1. LastMarkedFrameData: Temporarily keeps track of all the markers completed for a given frame thus far; resets when frame is completed. If the function is completed correctly, it should match the data contained in the final cell of the output (i.e. the data for the end frame).
1. CumFrameData: Temporarily keeps track of all the markers for all the frames completed thus far; ends when function ends. Thus, CumFrameData should look exactly like the output if the function is completed correctly.

```
function varargout = GUIcreateManualPoints(varargin)

% GUICREATEMANUALPOINTS MATLAB code for GUIcreateManualPoints.fig
%     GUICREATEMANUALPOINTS, by itself, creates a new GUICREATEMANUALPOINTS or ra
%     singleton*.
%
%     H = GUICREATEMANUALPOINTS returns the handle to a new GUICREATEMANUALPOINTS
%     the existing singleton*.
%
%     GUICREATEMANUALPOINTS('CALLBACK',hObject,eventData,handles,...) calls the l
%     function named CALLBACK in GUICREATEMANUALPOINTS.M with the given input arg
%
%     GUICREATEMANUALPOINTS('Property','Value',...) creates a new GUICREATEMANUAL
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before GUIcreateManualPoints_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to GUIcreateManualPoints_OpeningFcn via varargin
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUIcreateManualPoints

% Last Modified by GUIDE v2.5 14-May-2015 14:50:54

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @GUIcreateManualPoints_OpeningFcn, ...
    'gui_OutputFcn',  @GUIcreateManualPoints_OutputFcn, ...
    'gui_LayoutFcn',  [] , ...
    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
```

```
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

## Opening Function

Establishes variables used in later parts of function.

```
% --- Executes just before GUIcreateManualPoints is made visible.
```

```
function GUIcreateManualPoints_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
% This function has no output args, see OutputFcn.
```

```
% hObject    handle to figure
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles     structure with handles and user data (see GUIDATA)
```

```
% varargin    command line arguments to GUIcreateManualPoints (see % VARARGIN)
```

```
% Convert input variables into variables in function workspace. Determine
```

```
% end frame and import video to be analyzed
```

```
if nargin > 0;
```

```
    RatData = varargin{1};
```

```
    i = varargin{2};
```

```
    j = varargin{3};
```

```
    StartFrame = varargin{4};
```

```
end
```

```
EndFrame = StartFrame+40;
```

```
video = VideoReader(fullfile(RatData(i).DateFolders,RatData(i).VideoFiles(j).name)
```

```
% Create cell array of marker names to be displayed one-by-one in GUI
```

```
MarkerPoints = {'Pellet Center',...
    'Center of Back Surface of Paw',...
    'Far Index Finger Joint',...
    'Middle Index Finger Joint',...
    'Near Index Finger Joint',...
    'Far Middle Finger Joint',...
    'Middle Middle Finger Joint',...
    'Near Middle Finger Joint',...
    'Far Ring Finger Joint',...
    'Middle Ring Finger Joint',...
    'Near Ring Finger Joint',...
    'Far Pinky Finger Joint',...
    'Middle Pinky Finger Joint',...
    'Near Pinky Finger Joint'};
```

```
% Create cell array of anatomical marker names to be displayed with corresponding
```

```
MarkerPoints2 = {'',...
    '',...
    '(Metacarpal-Proximal Phalanges)',...
    '(Proximal-Middle Phalanges)',...
    '(Middle-Distal Phalanges)',...
    '(Metacarpal-Proximal Phalanges)',...
    '(Proximal-Middle Phalanges)',...
    ''};
```

```
'(Middle-Distal Phalanges)',...
'(Metacarpal-Proximal Phalanges)',...
'(Proximal-Middle Phalanges)',...
'(Middle-Distal Phalanges)',...
'(Metacarpal-Proximal Phalanges)',...
'(Proximal-Middle Phalanges)',...
'(Middle-Distal Phalanges)'};

% Cell array that will show in GUI which view the user should be placing
% markers in (left, center, or right)
FrameRegionInFocus = {'(Video) Left','Center','(Video) Right'};

% Add these variables created to the handles structure so the other
% callback functions for the various buttons and text boxes in the GUI can
% access it
handles.RatData = RatData;
handles.i = i;
handles.j = j;
handles.StartFrame = StartFrame;
handles.EndFrame = EndFrame;
handles.video = video;
handles.MarkerPoints = MarkerPoints;
handles.MarkerPoints2 = MarkerPoints2;
handles.FrameRegionInFocus = FrameRegionInFocus;

% Choose default command line output for GUIcreateManualPoints
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
% UIWAIT makes GUIcreateManualPoints wait for user response (see UIRESUME)
% This is part of what forces the function to wait until all the markers
% for all the frames have been placed before producing the output.
uiwait(handles.figure1);
```

## Output Function

Tells function to close the GUI and export AllFramesMarkerLocData (created in begin\_button\_Callback function below).

% --- Outputs from this function are returned to the command line.

```
function varargout = GUIcreateManualPoints_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
% varargout{1} = handles.output;
close(handles.figure1);
varargout{1} = handles.AllFramesMarkerLocData;
delete(handles.figure1);
```

```
close all;
```

*Attempt to reference field of non-structure array.*

*Error in GUIcreateManualPoints>GUIcreateManualPoints\_OutputFcn (line 193)*  
*close(handles.figure1);*

*Error in gui\_mainfcn (line 264)*  
*feval(gui\_State.gui\_OutputFcn, gui\_hFigure, [], gui\_Handles);*

*Error in GUIcreateManualPoints (line 95)*  
*gui\_mainfcn(gui\_State, varargin{:});*

## Irrelevant functions (with respect to user and GUI output)

correspond to the marker indicator text box

```
function marker_indicator_txtbox_Callback(hObject, eventdata, handles)
% hObject      handle to marker_indicator_txtbox (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of marker_indicator_txtbox as text
%         str2double(get(hObject,'String')) returns contents of marker_indicator_tx

% --- Executes during object creation, after setting all properties.
function marker_indicator_txtbox_CreateFcn(hObject, eventdata, handles)

% hObject      handle to marker_indicator_txtbox (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
    set(hObject,'BackgroundColor','white');
end
```

## Function executed upon pressing Begin button

Ultimately leads to creation of output, AllFramesMarkerLocData (see top)

```
% --- Executes on button press in begin_button.

function begin_button_Callback(hObject, eventdata, handles)

% hObject      handle to begin_button (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Import relevant variables from handles structure which were created in
```

Create Position Data for  
Joints of Rat's Paw to Deter-  
mine Fine Motor Movement

---

```
% opening function

StartFrame = handles.StartFrame;
EndFrame = handles.EndFrame;
video = handles.video;
MarkerPoints = handles.MarkerPoints;
MarkerPoints2 = handles.MarkerPoints2;
FrameRegionInFocus = handles.FrameRegionInFocus;

% Set interval for frame analysis after start frame. Create empty
% AllFramesMarkerLocData cell array
Interval = 10;
AllFramesMarkerLocData = cell(1,((EndFrame-StartFrame)./Interval));

% Create relevant substructures for AllFramesMarkerLocData.
% NOTE: IF A MARKER'S LOCATION HAS NOT BEEN RECORDED, IT WILL APPEAR AS '0'
% IN THE FINAL OUTPUT. IF THE USER INDICATES IT IS NOT VISIBLE IN A
% PARTICULAR REGION OF THE FRAME, IT WILL APPEAR AS 'NaN' IN OUTPUT. IF IT
% IS CORRECTLY RECORDED, IT'S COORDINATES WILL BE RECORDED BOTH IN THE
% FINAL OUTPUT AND THE "TEMPORARY" VARIABLES EXPLAINED BELOW

x = 0;
y = 0;
PelletCenter = table(x,y);
PawCenter = table(x,y);
x = zeros(4,1);
y = zeros(4,1);
ProximalPhalanges_MiddlePhalangesHulls = table(x,y,'RowNames',{'Index','Middle','R
MiddlePhalanges_DistalPhalangesHulls = table(x,y,'RowNames',{'Index','Middle','Rin
Metacarpal_ProximalPhalangesHulls = table(x,y,'RowNames',{'Index','Middle','Ring',

MarkerLocData = struct('PelletCenter', PelletCenter,...
    'PawCenter', PawCenter,...
    'MetacarpalPhalanges_ProximalPhalangesHulls', Metacarpal_ProximalPhalangesHull
    'ProximalPhalanges_MiddlePhalangesHulls', ProximalPhalanges_MiddlePhalangesHul
    'MiddlePhalanges_DistalPhalangesHulls', MiddlePhalanges_DistalPhalangesHulls);
% The code below is what controls marker placement.
r = 1;
% Starting with the start % frame and proceeding according to the interval
% set by Interval to the end frame ...
for k = StartFrame:Interval:EndFrame;
    % the function displays the frame image...
    im = read(video,k);
    figure;
    imshow(im);
    % creates the Frame Data substructure (with further substructures Left,
    % Center, and Right; corresponds to regions of frame under analysis)...
    FrameData = struct('Left',[],'Center',[],'Right',[]);
    for q = 1:3;
        n = 1;
        o = 1;
        p = 1;
        % displays in the GUI which region of the frame the user should be
        % marking...
```

Create Position Data for  
Joints of Rat's Paw to Deter-  
mine Fine Motor Movement

---

```
set(handles.frame_reg_in_focus_txtbox,'String',FrameRegionInFocus{q})
% proceeds through all the relevant markers...
for m = 1:length(MarkerPoints);
    % first displaying the marker's name and anatomical name in the
    % GUI...
    set(handles.marker_indicator_txtbox,'String',MarkerPoints{m});
    set(handles.marker_indicator2_txtbox,'String',MarkerPoints2{m});
    % then recording where the user clicks on the frame image to
    % indicate the marker (detailed instructions in GUI)...
    [x,y] = getpts;
    % for the pellet center marker...
    if m == 1;
        % if the user does not click and presses Enter (indicating
        % the marker is not visible), the function inserts a NaN
        % into the relevant table
        if isempty(x)||isempty(y);
            MarkerLocData.PelletCenter(1,1) = {NaN};
            MarkerLocData.PelletCenter(1,2) = {NaN};
        % otherwise, function records position data and displays a
        % BLUE circle temporarily where the user indicated the
        % pellet center is
        else
            MarkerLocData.PelletCenter(1,1) = num2cell(x);
            MarkerLocData.PelletCenter(1,2) = num2cell(y);
            PelletMarkerCircle = insertShape(im, 'FilledCircle', [x,y,8],
            imshow(PelletMarkerCircle);
        end
    % for the paw center marker...
    elseif m == 2;
        % if the user does not click and presses Enter (indicating
        % the marker is not visible), the function inserts a NaN
        % into the relevant table
        if isempty(x)||isempty(y);
            MarkerLocData.PawCenter(1,1) = {NaN};
            MarkerLocData.PawCenter(1,2) = {NaN};
        % otherwise, function records position data and displays a
        % RED circle temporarily where the user indicated the
        % paw center is
        else
            MarkerLocData.PawCenter(1,1) = num2cell(x);
            MarkerLocData.PawCenter(1,2) = num2cell(y);
            PawCenterMarkerCircle = insertShape(im, 'FilledCircle', [x,y,8],
            imshow(PawCenterMarkerCircle);
        end
    % for the metacarpal phalanges-proximal phalanges (McPh-pPh) joints...
    elseif m == 3 || m == 6 || m == 9 || m == 12;
        % if the user does not click and presses Enter (indicating
        % the marker is not visible), the function inserts a NaN
        % into the relevant table. It then increments so on the next itera
        % the joint for the next finger (in the order Index,
        % Middle, Ring, Pinky).
        if isempty(x)||isempty(y);
            MarkerLocData.MetacarpalPhalanges_ProximalPhalangesHulls(n,1)
            MarkerLocData.MetacarpalPhalanges_ProximalPhalangesHulls(n,2)
```

---

Create Position Data for  
Joints of Rat's Paw to Deter-  
mine Fine Motor Movement

---

```
n = n+1;
% otherwise, function records position data and displays a
% GREEN circle temporarily where the user indicated the
% McPh-pPh joint center is. It then increments so on the next iter
% the joint for the next finger (in the order Index,
% Middle, Ring, Pinky).
else
    MarkerLocData.MetacarpalPhalanges_ProximalPhalangesHulls(n,1)
    MarkerLocData.MetacarpalPhalanges_ProximalPhalangesHulls(n,2)
    McPh_pPhCenterMarkerCircle = insertShape(im, 'FilledCircle', [
    imshow(McPh_pPhCenterMarkerCircle);
    n = n+1;
end
% for the proximal phalanges-middle phalanges (pPh-mPh) joints...
elseif m == 4 || m == 7 || m == 10 || m == 13;
    % if the user does not click and presses Enter (indicating
    % the marker is not visible), the function inserts a NaN
    % into the relevant table. It then increments so on the next itera
    % the joint for the next finger (in the order Index,
    % Middle, Ring, Pinky).
    if isempty(x)||isempty(y);
        MarkerLocData.ProximalPhalanges_MiddlePhalangesHulls(o,1) = {NaN
        MarkerLocData.ProximalPhalanges_MiddlePhalangesHulls(o,2) = {NaN
        o = o+1;
    % otherwise, function records position data and displays a
    % cyan circle temporarily where the user indicated the
    % pPh-mPh joint center is. It then increments so on the next itera
    % the joint for the next finger (in the order Index,
    % Middle, Ring, Pinky).
    else
        MarkerLocData.ProximalPhalanges_MiddlePhalangesHulls(o,1) = num2
        MarkerLocData.ProximalPhalanges_MiddlePhalangesHulls(o,2) = num2
        pPh_mPhCenterMarkerCircle = insertShape(im, 'FilledCircle', [x
        imshow(pPh_mPhCenterMarkerCircle);
        o = o+1;
    end
% for the middle phalanges-distal phalanges (mPh-dPh) joints...
elseif m == 5 || m == 8 || m == 11 || m == 14;
    % if the user does not click and presses Enter (indicating
    % the marker is not visible), the function inserts a NaN
    % into the relevant table. It then increments so on the next itera
    % the joint for the next finger (in the order Index,
    % Middle, Ring, Pinky).
    if isempty(x)||isempty(y);
        MarkerLocData.MiddlePhalanges_DistalPhalangesHulls(p,1) = {NaN
        MarkerLocData.MiddlePhalanges_DistalPhalangesHulls(p,2) = {NaN
        p = p+1;
    % otherwise, function records position data and displays a
    % magenta circle temporarily where the user indicated the
    % mPh-dPh joint center is. It then increments so on the next itera
    % the joint for the next finger (in the order Index,
    % Middle, Ring, Pinky).
    else
        MarkerLocData.MiddlePhalanges_DistalPhalangesHulls(p,1) = num2
```



Create Position Data for  
Joints of Rat's Paw to Deter-  
mine Fine Motor Movement

---

```

MarkerLocData.MiddlePhalanges_DistalPhalangesHulls(p,2) = num2
mPh_dPhCenterMarkerCircle = insertShape(im, 'FilledCircle', [x
imshow(mPh_dPhCenterMarkerCircle);
p = p+1;
end
end
% the last marked marker position data is exported to the base
% workspace in the structure LastMarkedMarkerData, which builds
% until it is filled for a given frame region, then resets
assignin('base','LastMarkedMarkerLocData', MarkerLocData);
end
% the frame region marker data is put in the appropriate structure
if q == 1;
    FrameData.Left = MarkerLocData;
elseif q == 2;
    FrameData.Center = MarkerLocData;
elseif q == 3;
    FrameData.Right = MarkerLocData;
end
% All the markers for a given frame region for a given frame are exported
% workspace in the structure LastMarkedFrameData, which resets when
% a frame is completed.
assignin('base','LastMarkedFrameData', FrameData);
end
% the total frame data for all regions is put into the output structure
% AllFramesMarkerLocData
AllFramesMarkerLocData{r} = FrameData;
% After a frame is done, all of its data is exported to the base
% workspace in the structure CumFrameData, which builds until all
% frames are completed. So, ideally, when the program is completed,
% CumFrameData should look exactly like the output
% AllFramesMarkerLocData. If it is ended prematurely, it saves up to
% the last completed frame.
assignin('base','CumFrameData',AllFramesMarkerLocData);
r = r+1;
handles.k = k;
% output (AllFramesMarkerLocData) updated in handles structure with
% every frame completion
handles.AllFramesMarkerLocData = AllFramesMarkerLocData;
guidata(hObject, handles);
close;
end
% Once all frames are completed, proceed to output function by allowing the
% GUI to resume
uiresume(handles.figure1);

Warning: Image is too big to fit on screen;
displaying at 67%
Warning: Image is too big to fit on screen;
displaying at 67%
Error using getpts (line 173)
Interruption during mouse point selection.

Error in

```

---

Create Position Data for  
Joints of Rat's Paw to Determine Fine Motor Movement

---

```
GUIcreateManualPoints>begin_button_Callback  
(line 294)
```

```
    [x,y] = getpts;
```

```
Error in gui_mainfcn (line 95)  
    feval(varargin{:});
```

```
Error in GUIcreateManualPoints (line 95)  
    gui_mainfcn(gui_State, varargin{:});
```

```
Error in
```

```
@(hObject,eventdata)GUIcreateManualPoints('begin_button_Callback',hObject,eventdata)
```

```
Error using waitfor
```

```
Error while evaluating UIControl Callback
```

## More irrelevant functions (with respect to user and GUI output)

```
% --- Executes during object creation, after setting all properties.
```

```
function frame_display_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to frame_display (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: place code in OpeningFcn to populate frame_display
```

```
function frame_reg_in_focus_txtbox_Callback(hObject, eventdata, handles)  
% hObject    handle to frame_reg_in_focus_txtbox (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of frame_reg_in_focus_txtbox as text  
%        str2double(get(hObject,'String')) returns contents of frame_reg_in_focus_txtbox as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function frame_reg_in_focus_txtbox_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to frame_reg_in_focus_txtbox (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%        See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundcolor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function marker_indicator2_txtbox_Callback(hObject, eventdata, handles)
% hObject      handle to marker_indicator2_txtbox (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of marker_indicator2_txtbox as text
%         str2double(get(hObject,'String')) returns contents of marker_indicator2_t

% --- Executes during object creation, after setting all properties.
function marker_indicator2_txtbox_CreateFcn(hObject, eventdata, handles)
% hObject      handle to marker_indicator2_txtbox (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
    set(hObject,'BackgroundColor','white');
end
```

*Published with MATLAB® R2014b*