

---

## Table of Contents

.....	1
Import data .....	1
Preprocess gaze data .....	1
Plot outcome .....	1
Process gaze data .....	3
Load trial data .....	6

```
clear
close all
```

## Import data

```
% fn = 'EyeTracker\SurfaceTest3.p.mat';
fn = 'Data\7\23-Jan-2017_15_16_25\7_ProcessedGaze.mat';
[gaze, nG] = loadGaze(fn, {'TS', 'NP0', 'NP1', 'onSurf'});
```

## Preprocess gaze data

```
% Remove empty rows
gaze = gaze(~isnan(gaze.onSurf),:);

% % Apply threshold
% gazeThresh = 1;
% idx = gaze.onSurf>=gazeThresh;
% gaze.onSurf(idx) = true;
% gaze.onSurf(~idx) = false;

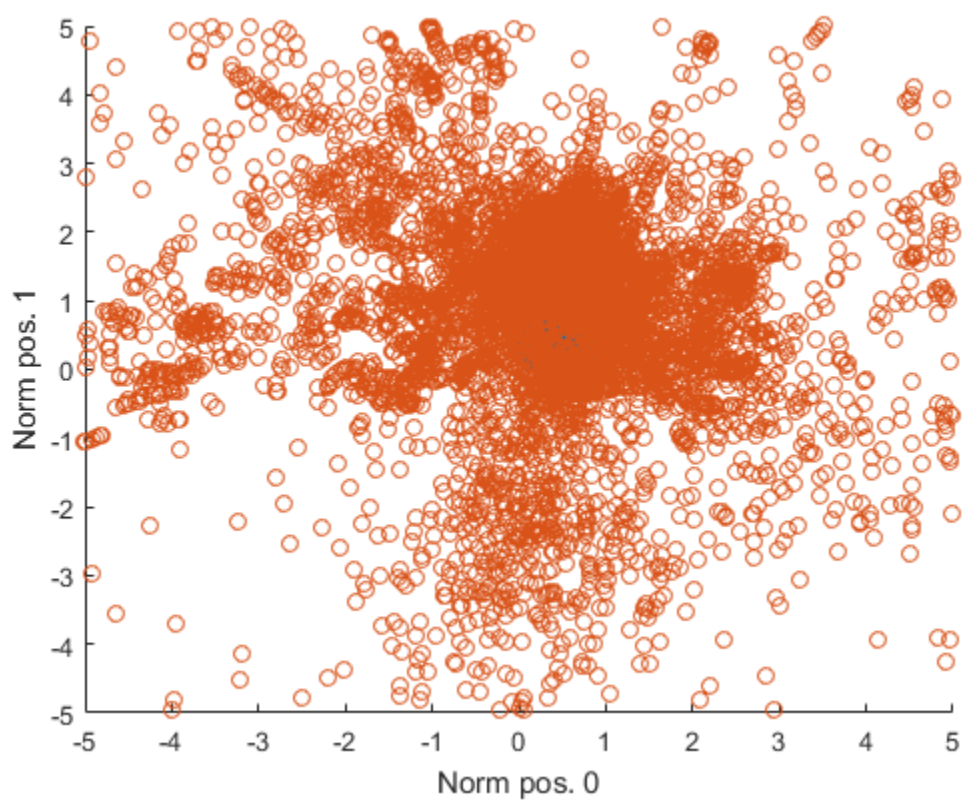
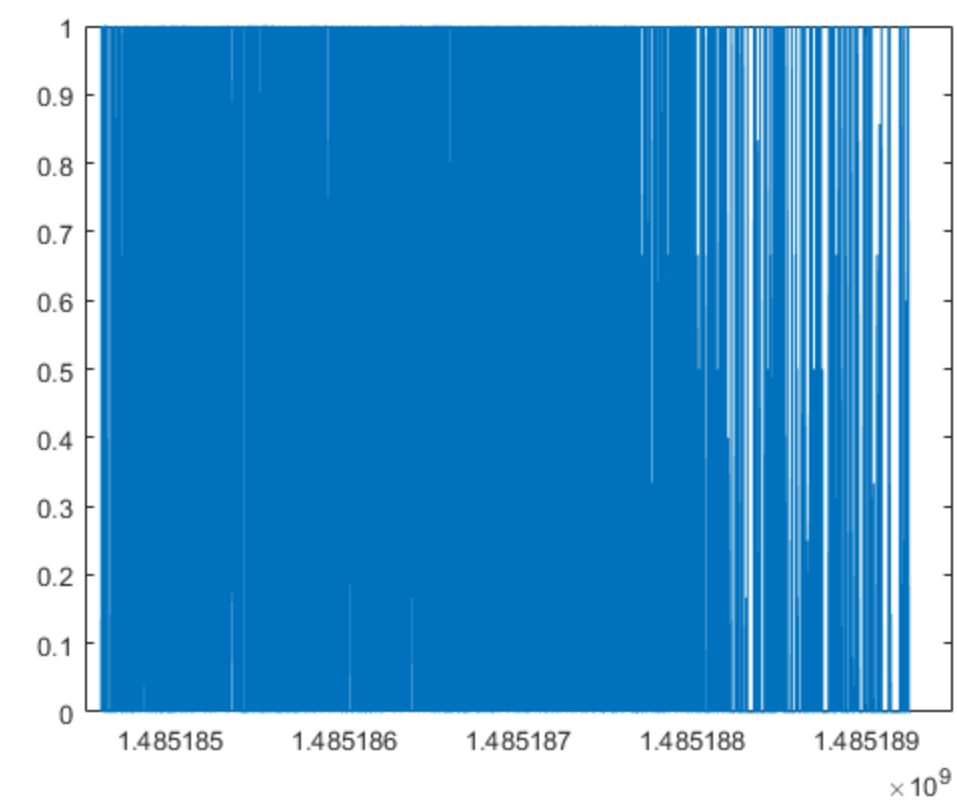
% Convert time
gaze.TS2 = datetime(gaze.TS, 'ConvertFrom', 'posixtime');
```

## Plot outcome

```
close all

figure
plot(gaze.TS, gaze.onSurf)

plotGaze([gaze.NP0, gaze.NP1], gaze.onSurf, 'RawGaze')
```



---

# Process gaze data

Centre space, reduce noise, correct for drift during experiment. Expand surface?

```
% Steps:
% Find surface region in gaze
% Copy gaze to gazeCorrected
% Limit extreme values of NP
% Plot (sp 1 of 3)
% Find and correct for drift. This zeros space.
% Limit new extreme values
% Plot (sp 2 and 3 of 3)
% Shift surface to centre of space
% Expand square surface by some factor
% Recalculate onSurf
% Calcualte "surface" from eculidan distance from [0,0]
(gazeCorrected.onSurfED)
% Plot gazeCorrected.onSurf
% Plot gazeCorrected.onSurfED

% Parameters
lim = 4; % Extreme value limit
sExp = 1.5; % Surface expansion factor

gazeCorrected = gaze;

% onSurf extent [xMin, xMax, yMin, yMax]
onSurfEx = [min(gaze.NP0(gaze.onSurf==true)), ...
max(gaze.NP0(gaze.onSurf==true)), ...
min(gaze.NP1(gaze.onSurf==true)), ...
max(gaze.NP1(gaze.onSurf==true))];

% First, cap extreme values
gazeCorrected.NP0(gazeCorrected.NP0>lim) = lim;
gazeCorrected.NP0(gazeCorrected.NP0<-lim) = -lim;
gazeCorrected.NP1(gazeCorrected.NP1>lim) = lim;
gazeCorrected.NP1(gazeCorrected.NP1<-lim) = -lim;

subplot(3,1,1)
plot([gazeCorrected.NP0, gazeCorrected.NP1])
ylim([-5,5])
title('Limited gaze')

% Find drift in eye data
% Does mean eye position drift with time?
mAvRange = 2000; % pts
drift.NP0 = tsmovavg(gazeCorrected.NP0, 's', mAvRange, 1);
drift.NP1 = tsmovavg(gazeCorrected.NP1, 's', mAvRange, 1);

% Corret drift in NP0
gazeCorrected.NP0(mAvRange+1:end) = ...
    gaze.NP0(mAvRange+1:end) ...
    - drift.NP0(mAvRange+1:end); % Avoid adding NaNs
```

---

```

% Corret drift in NP1
gazeCorrected.NP1(mAvRange+1:end) = ...
    gaze.NP1(mAvRange+1:end) ...
    - drift.NP1(mAvRange+1:end); % Avoid adding NaNs

% Again, limit extreme values
gazeCorrected.NP0(gazeCorrected.NP0>lim) = lim;
gazeCorrected.NP0(gazeCorrected.NP0<-lim) = -lim;
gazeCorrected.NP1(gazeCorrected.NP1>lim) = lim;
gazeCorrected.NP1(gazeCorrected.NP1<-lim) = -lim;

% Finish current figure
subplot(3,1,2)
plot([drift.NP0, drift.NP1])
ylim([-5,5])
title('Gaze drift')

subplot(3,1,3)
plot([gazeCorrected.NP0, gazeCorrected.NP1])
ylim([-5,5])
title('Corrected gaze')

% Space has now been centered around [0,0] (hopefully)
% Shift surface from eg. [0,1,0,1] to [-0.5,0.5,-0.5,0.5]
onSurfEx(1:2) = onSurfEx(1:2) - mean(onSurfEx(1:2));
onSurfEx(3:4) = onSurfEx(3:4) - mean(onSurfEx(3:4));
% Also expand by some tolerance value
onSurfEx = onSurfEx*sExp;

% Reclassify onSurf
gazeCorrected.onSurf = ...
    gazeCorrected.NP0>onSurfEx(1) ...
    & gazeCorrected.NP0<onSurfEx(2) ...
    & gazeCorrected.NP1>onSurfEx(3) ...
    & gazeCorrected.NP1<onSurfEx(4);

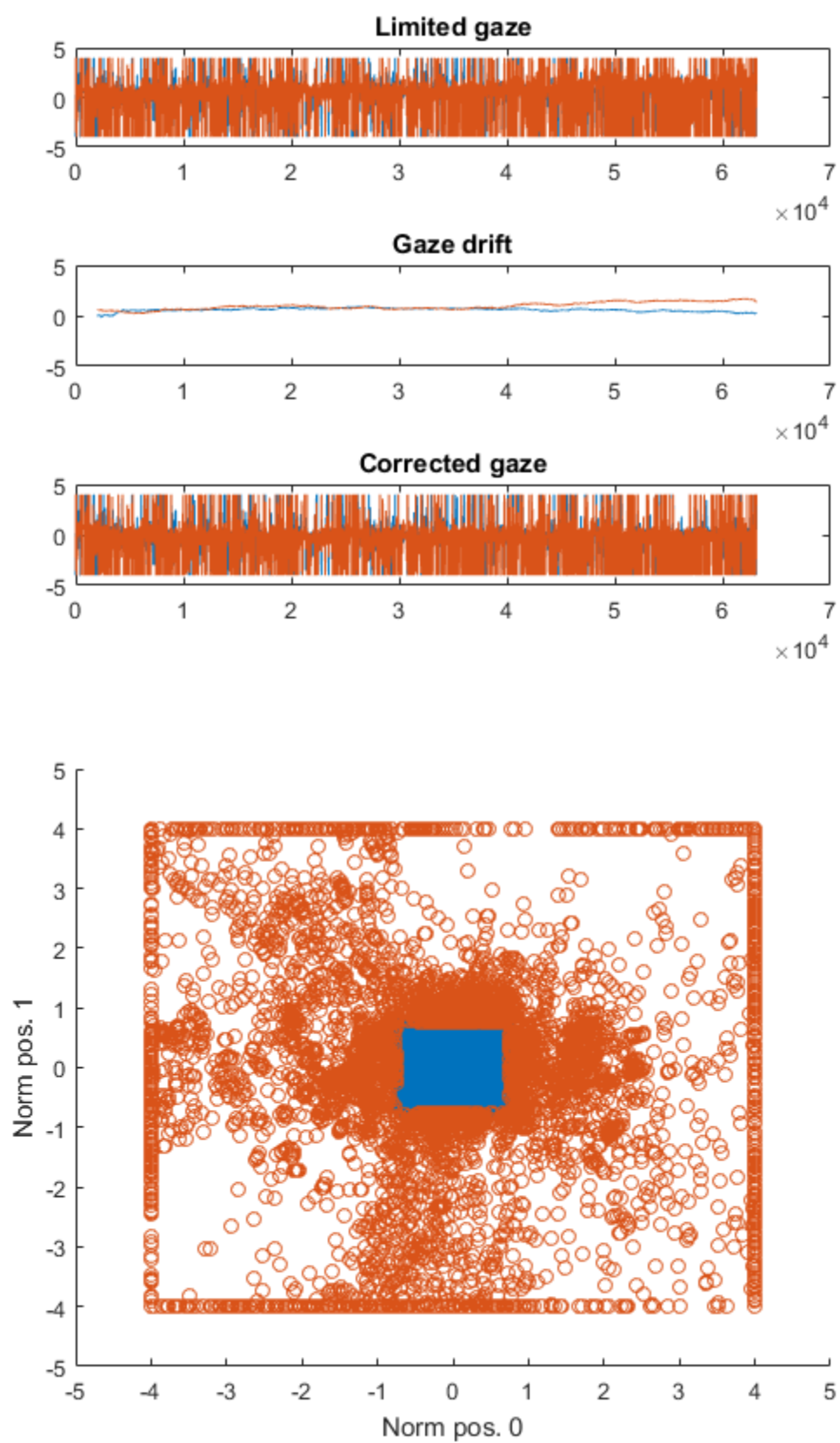
% Try eculidian distance
gazeCorrected.ED = ...
    sqrt(gazeCorrected.NP0.^2 + gazeCorrected.NP1.^2);
EDLim = max(onSurfEx*sExp);
gazeCorrected.onSurfED = gazeCorrected.ED<EDLim;

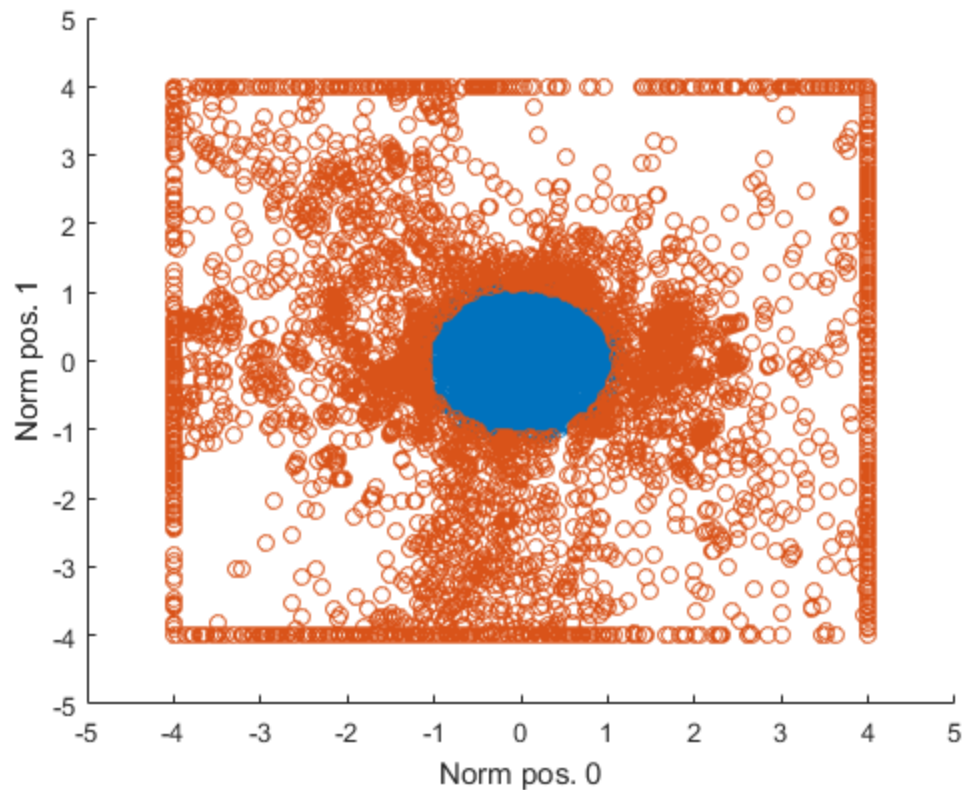
% Plot onSurf comparison
plotGaze([gazeCorrected.NP0, gazeCorrected.NP1], ...
    gazeCorrected.onSurf, 'gazeCorrected')

% Plot onSurf comparison (ED)
plotGaze([gazeCorrected.NP0, gazeCorrected.NP1], ...
    gazeCorrected.onSurfED, 'gazeCorrected - ED')

```

---





## Load trial data

```
gazePropThresh = 0.1;

fn = 'Data\7\23-Jan-2017 15_16_25\SpatialCapture_7.mat';
a = load(fn);
stimLog = a.stimLog(~isnan(a.stimLog.PosBin(:,1)),:);
nT = height(stimLog);

% Convert time
stimLog.sTime = datetime(stimLog.startClock);
stimLog.eTime = datetime(stimLog.endClock);

figure
hold on
% Add gaze data to stimLog
stimLog.onSurf = NaN(nT,1);
stimLog.nGazeSamples = NaN(nT,1);
stimLog.onSurfProp = NaN(nT,1);
for r = 1:nT
    ts = stimLog.sTime(r);
    te = stimLog.eTime(r);

    tIdx = gazeCorrected.TS2>=ts & gazeCorrected.TS2<=te;
```

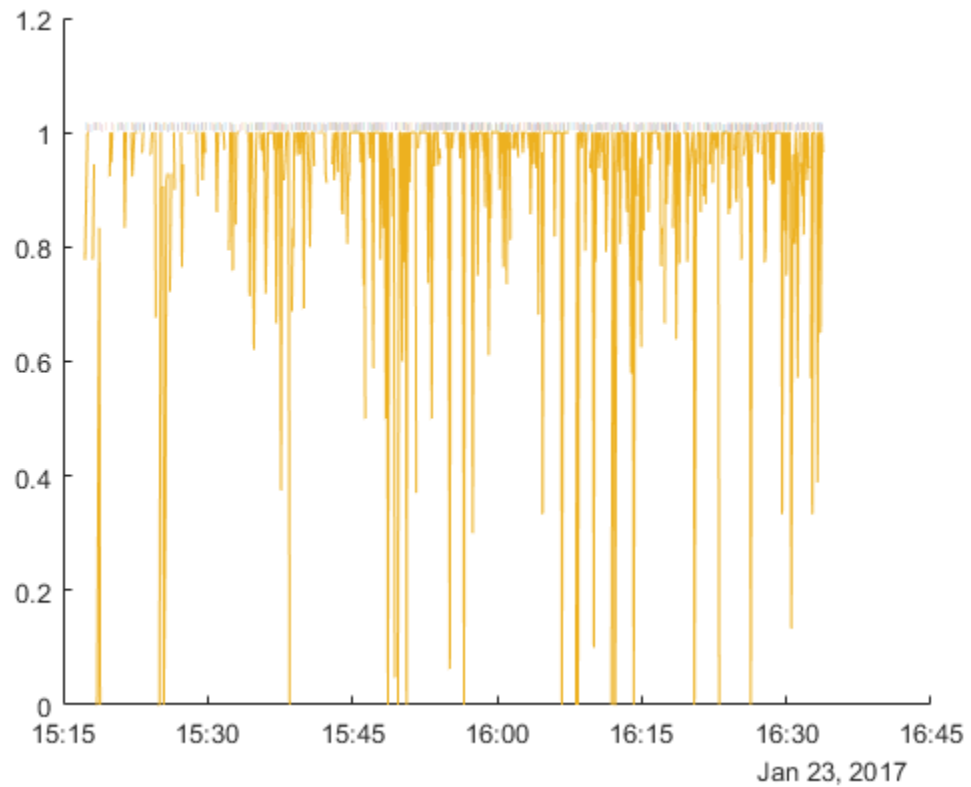
---

```
gs = gazeCorrected.onSurfED(tIdx);

stimLog.nGazeSamples(r) = numel(gs);
stimLog.onSurfProp(r) = mean(gs);

plot([ts,te], [1.01, 1.01], 'LineWidth', 3)
end

plot(stimLog.sTime, stimLog.onSurfProp)
% plot(stimLog.sTime, stimLog.onSurfProp>=gazePropThresh)
```



*Published with MATLAB® R2016b*