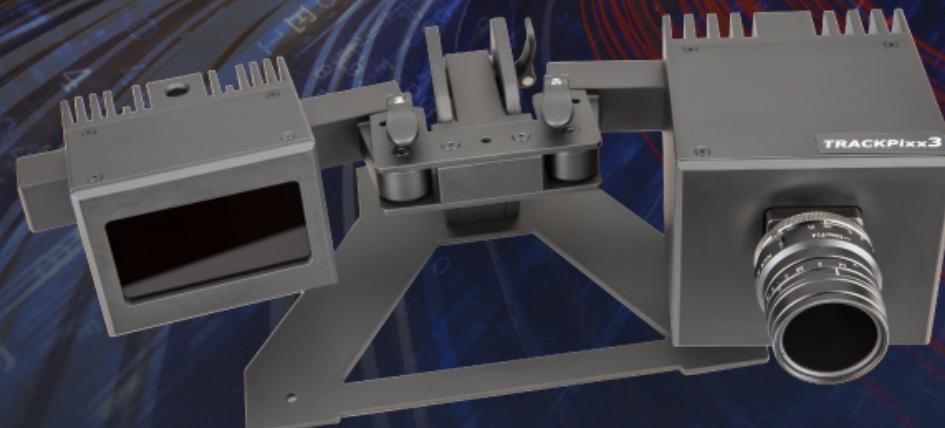


How to: TrackPixx3

**2 kHz binocular
Eye Tracker**

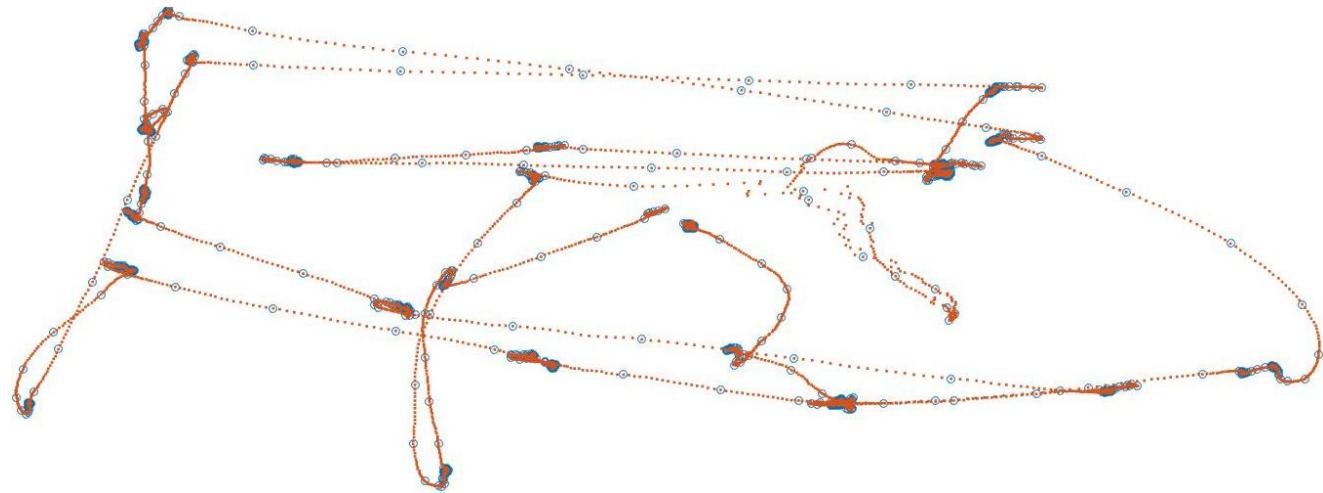
VideoBahn™
The NEW fast road to SUPERIOR data



Richard Schweitzer, 28.03.2019

How to: TrackPixx3

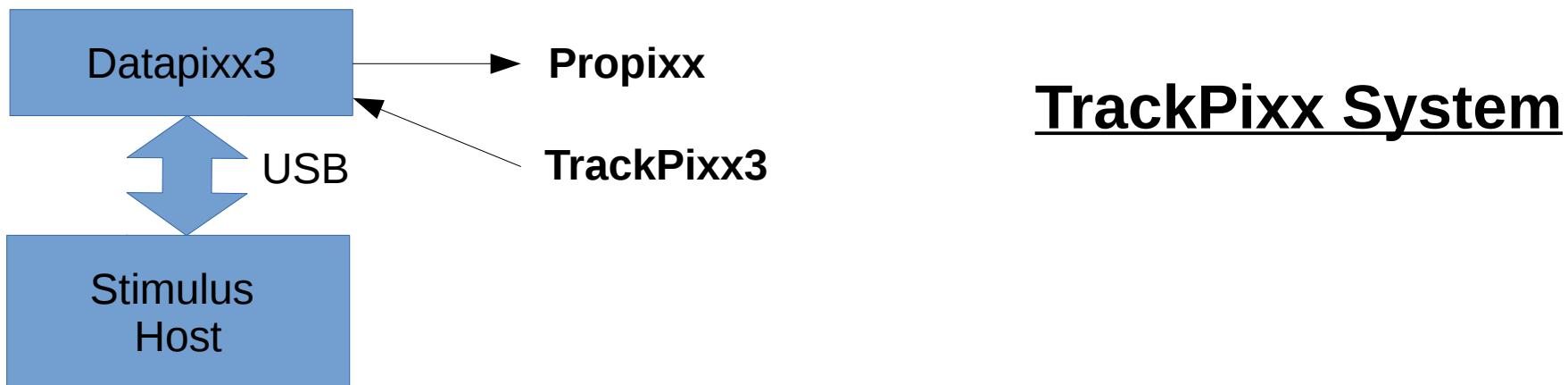
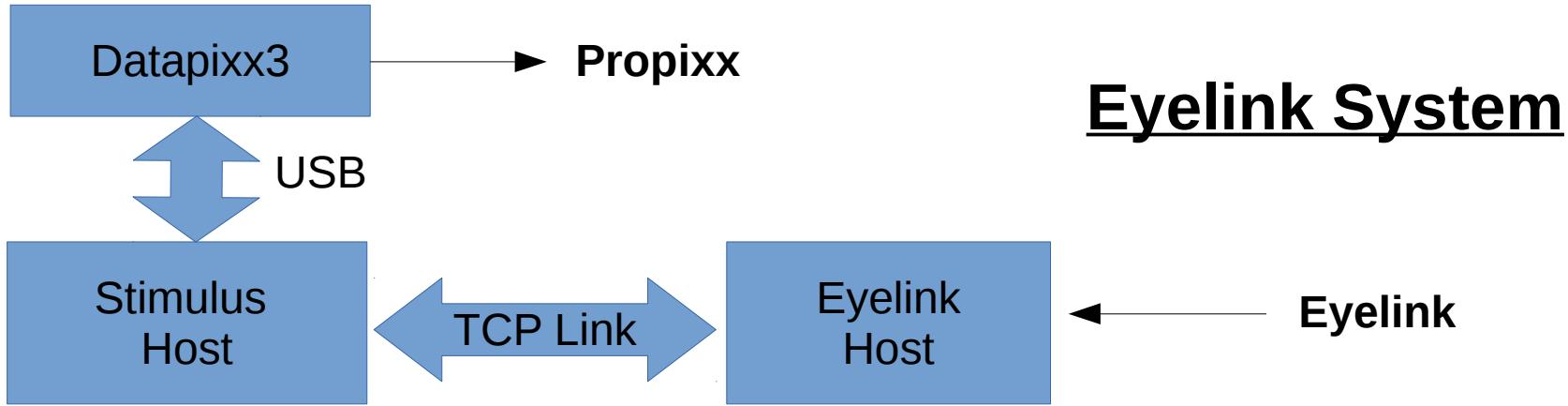
- What, why?
 - Specs
 - Setup
 - Pros and cons
- How?
 - Calibration / Validation
 - Data recording and retrieval
 - **Code:** Minimal example



Specs

	Eyelink 2	Eyelink 1000+	TrackPixx3
<i>Sampling rate</i>	250, 500 Hz	250 - 2000 Hz	2000 Hz
<i>Delay</i>	3-7 ms / 6-14 ms	1.34 – 6 ms	1.95 ms (constant)
<i>Method</i>	Pupil / Pupil-CR	Pupil-CR	Pupil-CR
<i>Accuracy</i>	0.5°	0.15 – 0.5°	0.2 – 0.6°
<i>Noise (RMS)</i>	< 0.01° / < 0.02°	0.01 – 0.05°	0.04°
<i>Resolution</i>	< 0.005°	0.01°	0.01°
<i>Pupil size resolution</i>	0.1%	0.1%	+/- 0.1 mm

Setup



Pros and cons

	Eyelink system	TrackPixed system
<i>Setup</i>	<ul style="list-style-type: none">- separate PC, screen, power necessary+ great experimenter view with real-time gaze position and AOI	<ul style="list-style-type: none">+ no need for an eye tracker host, control via Datapixx toolbox- hardly any experimenter features, nor experimenter interface (yet)
<i>Data</i>	<ul style="list-style-type: none">- messaging necessary (TCP and processing delays)- .edf files (incl. parsing...)+ automatic event messages- piece-wise online sample retrieval at variable latencies	<ul style="list-style-type: none">+ near perfect temporal synchronization (via PsychDatapixx)+ no separate data files- no event detection (yet)+ full online access to all samples at low, deterministic latency
<i>Handling</i>	<ul style="list-style-type: none">+ advanced, well documented Eyelink toolbox (Cornelissen et al.)- rigid, proprietary routines	<ul style="list-style-type: none">- all functions are beta, unpredictable updates, hardly any documentation+ more flexible adjustments possible

How to run the TrackPixx:

% TRACKPixx (any kind) Functions:

- Datapixx('GetEyeDuringCalibration', xScreen, yScreen);
- [xRawRight yRawRight xRawLeft yRawLeft] = Datapixx('GetEyeDuringCalibrationRaw', xScreen, yScreen);
- Datapixx('FinishCalibration');
- calibrations_coeff = Datapixx('GetCalibrationCoeff');
- [xScreenRight yScreenRight xScreenLeft yScreenLeft xRawRight yRawRight xRawLeft yRawLeft timetag] = Datapixx('GetEyePosition');
- convertedArray = Datapixx('ConvertCoordSysToCartesian', sourceArray, offsetX, scaleX, offsetY, scaleY);
- convertedArray = Datapixx('ConvertCoordSysToCustom', sourceArray, offsetX, scaleX, offsetY, scaleY);
- Datapixx('SetTPxAwake');
- Datapixx('EnableSearchLimits');
- Datapixx('DisableSearchLimits');
- Datapixx('ClearSearchLimits');
- Datapixx('SetSearchLimits', leftEye, rightEye);
- [leftEye, rightEye] = Datapixx('GetSearchLimits');
- Datapixx('EnableTrackpixxAnalogOutput', [eyeNumber=0])
- Datapixx('DisableTrackpixxAnalogOutput');
- fov_h = Datapixx('GetHorizontalFOV');
- fov_v = Datapixx('GetVerticalFOV');

% TRACKPixx3 only Functions:

- Datapixx('SaveCalibration');
- Datapixx('LoadCalibration');
- Datapixx('ClearCalibration');
- image = Datapixx('GetEyeImage');
- Datapixx('SetLedIntensity', ledIntensity);
- ledIntensity = Datapixx('GetLedIntensity');
- Datapixx('SetExpectedIrisSizeInPixels', IrisSize);
- expectedIrisSize = Datapixx('GetExpectedIrisSizeInPixels');
- pupilSize = Datapixx('GetPupilSizeSimple');
- [ppLeftMajor ppLeftMinor ppRightMajor ppRightMinor] = Datapixx('GetPupilSize');
- [ppLeftX ppLeftY ppRightX ppRightY] = Datapixx('GetPupilCoordinatesInPixels');
- [CRLefX CRLeftY CRRightX CRRightY] = Datapixx('GetCRCordinatesInPixels');
- Datapixx('SetupTPxSchedule', [bufferbaseAddress=12e6, numberofEyeData=60000]);
- Datapixx('StopTPxSchedule');
- Datapixx('StartTPxSchedule');
- [bufferData, underflow, overflow] = Datapixx('ReadTPxData', numFrames);
- status = Datapixx('GetTPxStatus');
- **AND MANY MORE!**

The TrackPixx Toolbox

- Goal: Comprehensible, easy-to-use toolbox
- Not yet public, but available for you:
<https://github.com/rolfslab/TrackPixxToolbox>
- Features:
 - Calibration / Validation
 - Initialize, start and stop recordings
 - Sample-wise and buffer-based (online) data retrieval
 - Demo (gaze-contingent, all modes, with timing)

TPx Calibration

MATLAB R2015a - academic use

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Compare Go To Insert Comment % fx Breakpoints Run Run and Advance Run Run Time

FILE NAVIGATE EDIT BREAKPOINTS RUN

Current Folder Editor - /local/locallab/Documents/Projects/richard/Trackpix Setup/try_trackpix.m Workspace Command Window

Editor - /local/locallab/Documents/Projects/richard/Trackpix Setup/try_trackpix.m

```
%> %% global parameters for recording and display
% use 1440_mode = 0; % not implemented yet
led_intensity = 8;
iris_size_pix = 140;
record_for_time = 60; % recording time in seconds
extra_time = 10; % extra time for pre-allocation
online_buffer_retrieval = 0; % how to retrieve data online? 0: 'GetEyePosition', 1: 'ReadTPxData'
assert(online_buffer_retrieval==0 || online_buffer_retrieval==1);

% screen parameters
%
% This is read from the BUFFER:
% timetag : bufferData(:,1)
% left_raw_x : bufferData(:,2)
% left_raw_y : bufferData(:,3)
% leftP_hori : bufferData(:,4)
% leftP_vert : bufferData(:,5)
% left_angle : bufferData(:,6)
% right_raw_x : bufferData(:,7)
% right_raw_y : bufferData(:,8)
% rightP_hori : bufferData(:,9)
% rightP_vert : bufferData(:,10)
% right_angle : bufferData(:,11)
% Digital Input values : bufferData(:,12)
% blink_left : bufferData(:,13)
% blink_right : bufferData(:,14)

%% initialize datapixx and trackpixx
Datapixx('Uninitialize');
WaitSecs(1);
Datapixx('Initialize', 0);
Datapixx('Open');
Datapixx('HideOverlay');
Datapixx('RegwRd');
Datapixx('SetTpxAwake');
Datapixx('RegwRd');
Datapixx('SetLedIntensity', led_intensity);
Datapixx('SetExpectedIrisSizeInPixels', iris_size_pix)
Datapixx('RegwRd');

first_eye_image = Datapixx('GetEyeImage');
%imshow(first_eye_image)

%% Setup Screen etc
thisScreen = max(Screen('Screens'));
scrGray = GrayIndex(thisScreen);
scrBlack = BlackIndex(thisScreen);
```

Workspace

Name	Type	Value	Max
ans	double	1	1
buffer_leftEyeScreen	double	10834x2 double	1491
buffer_readout_duration	double	12.1870	12.1870
buffer_readout_end	double	1.5393e+09	1.5393e+09
buffer_readout_start	double	1.5393e+09	1.5393e+09
buffer_rightEyeScreen	double	10834x2 double	1491
bufferData	double	10834x19 double	15393e+09
bufferSize	double	140000	140000
convert_durations	double	140000x1 double	NaN
current_buffer_size	double	14	14
currentWriteFrame	double	10834	10834
extra_time	double	10	10
finish_time	double	3.0859e+04	3.0859e+04
first_calib_res	struct	1x1 struct	<Too many levels>
first_eye_image	uint8	1280x1024 uint8	<Too many levels>
flip	double	609	609
flip_times	double	84000x1 double	NaN
im	uint8	1125x2000x3 uint8	<Too many levels>
im_texture	double	11	11
iris_size_pix	double	140	140
iris_size_px	double	140000	140000

Command Window

```
numStreamOverflows: 0

Return data initiated, about to read!
Buffer readout took 12.19 millisec.
With 10834 samples, this means 0.00112 millisec per sample.
Setting LED intensity to (0)

INFO: PTB's Screen('Flip', 10) command seems to have missed 1
INFO: a total of 6 times out of a total of 3669 flips during

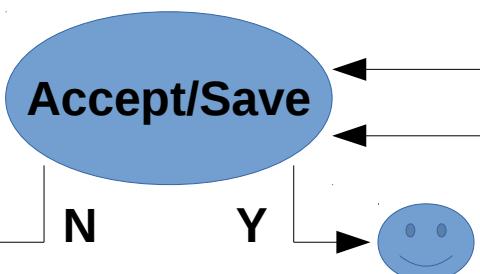
INFO: This number is fairly accurate (and indicative of real
INFO: if you provided requested stimulus onset times with the
INFO: If you called Screen('Flip', window); without the 'when'
INFO: of timing behaviour than a hard reliable measurement. I
INFO: deserve your closer attention. Cfr. 'Help SyncTrouble'.
INFO: examples in the PDF presentation in PsychDocumentation

percent_dropped =

1.1905e-04
```

f1>> try_trackpixx

1. Setup Camera



2. Selection of search limits

- Select eyes on screen:
 - left eye by **left click**
 - right eye by **right click**
- In-/decrease size of search limits: (+) **D**, (-) **A**
- Clear search limits: **C**
- Initiate calibration:
 - Auto mode: **Enter**
 - Manual mode: **M**

TPx Calibration

5b. Validation

5a. Gaze follower

3. Calibration

Parameters:

- background color
- screen PPD
- number of points
- x, y eccentricity
- highlight duration
- sampling duration
- LED intensity
- Iris size

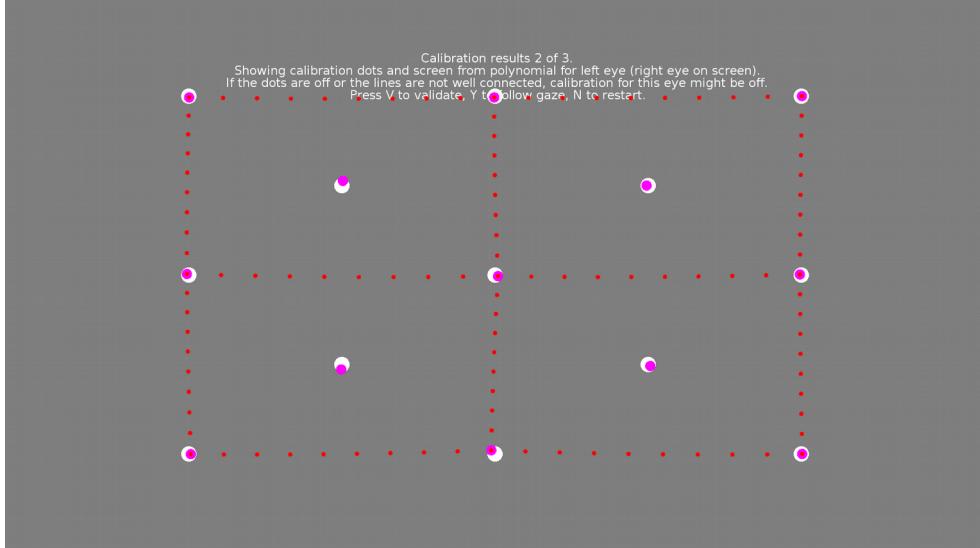
4. Review results

- Check number of points and relative distances
- Check the each polynomial fit
- Repeat (**N**), start Validation (**V**) or Gaze Follower (**Y**)

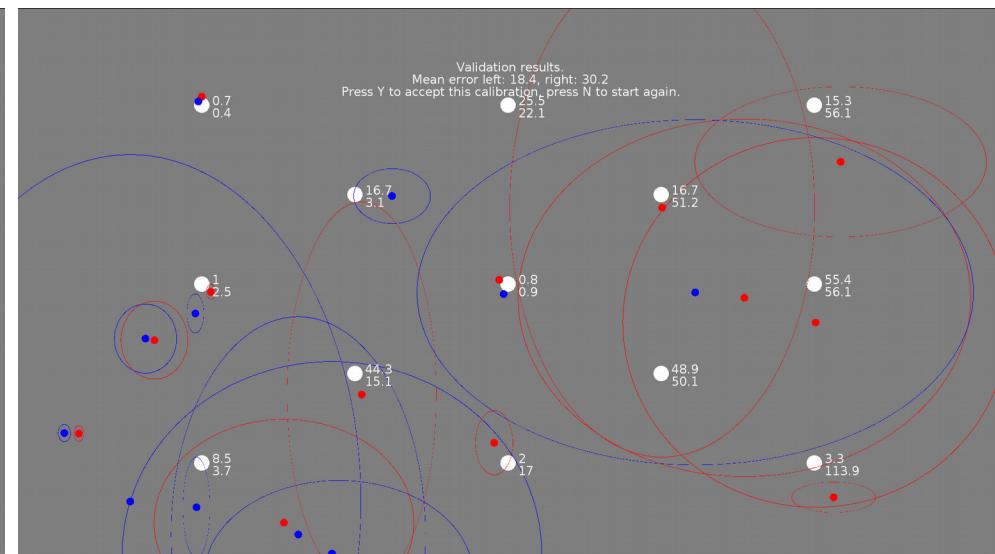
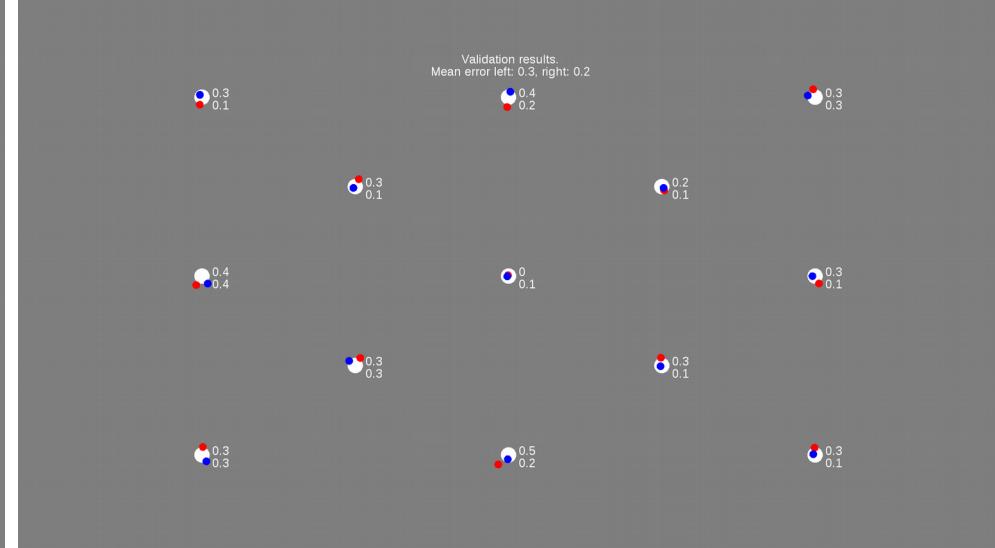
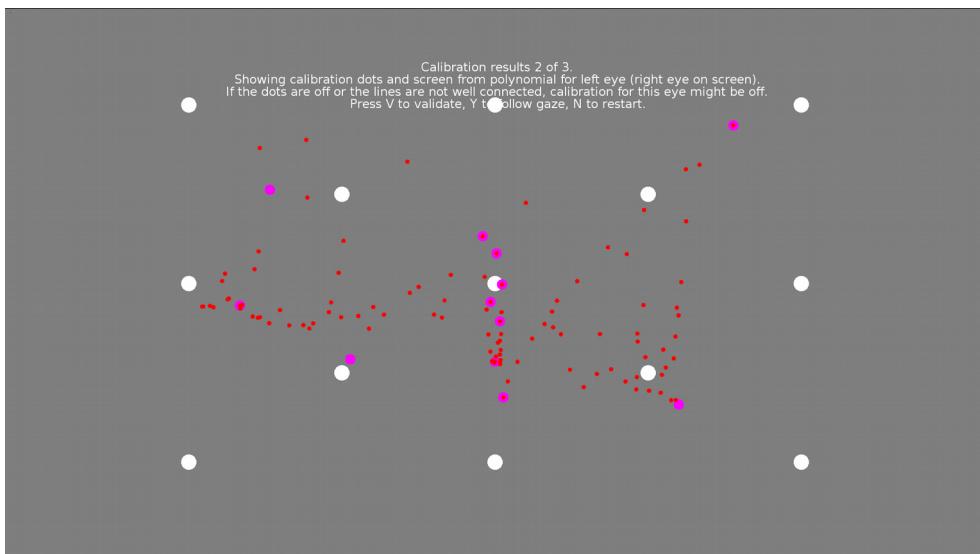
Calibration

Validation

Good



Bad



TPx Recording

- TrackPixed recordings based on a buffer allocated on the Datapixx3
- But: buffer is not a necessary condition!
- “default is 3593750 frames, or 30 minutes at 2KHz of recording”
- Multiple data retrievals from buffer possible
→ Online access to all data in the buffer

TPx Recording

- (nearly) all values of interest are available ...

	time_DP	right_raw_x	right_raw_y	right_pupil_x	right_pupil_y	right_angle
1	464400.0	77	-91	46.28906	45.32812	0.3750000
2	464400.5	77	-91	46.30469	45.32031	0.4687500
3	464401.0	77	-91	46.30078	45.32812	0.5195312

	left_raw_x	left_raw_y	left_pupil_x	left_pupil_y	left_angle	digital_in
	118	-67	44.95703	42.55859	0.4062500	65535
	117	-67	44.98438	42.55859	0.4453125	65535
	118	-67	44.96094	42.56641	0.4531250	65535

	right_blink	left_blink	time_pc_fast	right_x	right_y	left_x	left_y	time_pc
	0	0	1.553268e+12	518.5	315.5	539.0	303.5	1.553268e+12
	0	0	1.553268e+12	518.5	315.5	538.5	303.5	1.553268e+12
	0	0	1.553268e+12	518.5	315.5	539.0	303.5	1.553268e+12

Code: Initialize Recording

```
led_intensity = 8; % strength of illuminator [0..8]
iris_size_pix = 140; % Vpixx constant for specific lens
% initialize datapixx and trackpixx
Datapixx('Open');
InitializeTPx(led_intensity, iris_size_pix);
% open a psychtoolbox screen (you know the drill)
>windowPtr, windowRect] = PsychImaging('OpenWindow',...
    0, 0.5);
```

Code: Calibrate & Start recording

```
first_calib_res = doTPxCalibration(windowPtr, 0,...  
scr.gray, scr.ppd); % run default calibration and save result  
Screen('Flip', windowPtr);  
KbReleaseWait;  
% start recording with allocated buffer duration: record_for_sec  
[rec_start_time_pc, rec_start_time, ...  
bufferData, sample_i] = StartTPxRecording(record_for_sec);  
bufferData = [ ]; % in case we don't want pre-allocation
```

Code: Online Data Retrieval (1)

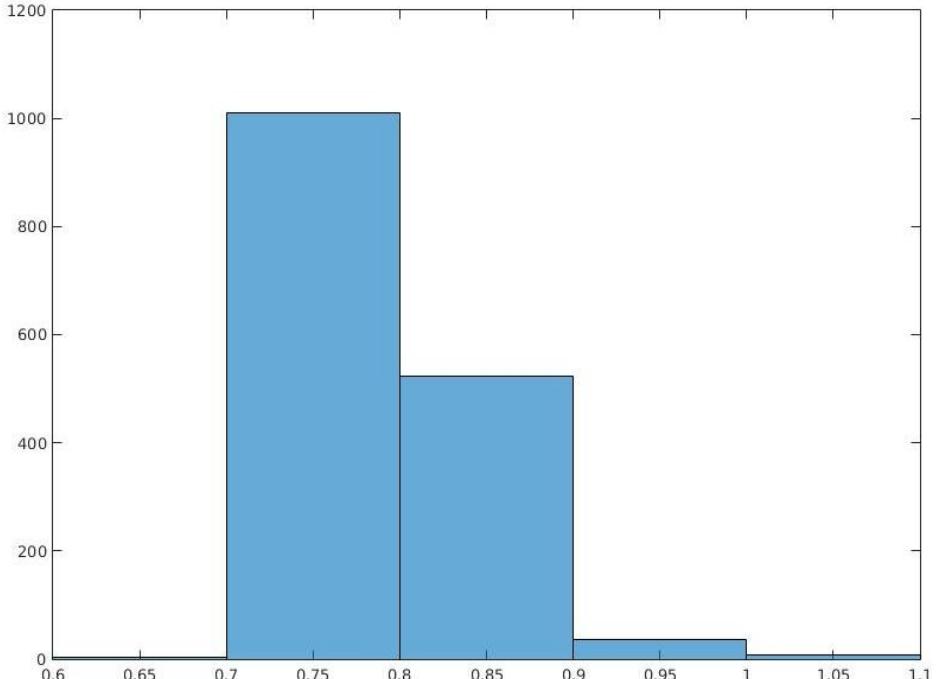
```
% buffer-based online data retrieval (no pre-allocation, intended)
[sample_i, lastRetrieval] = ...
    ReadTPxDataSimple(sample_i);
bufferData = [bufferData; lastRetrieval];
% compute 'current gaze position' based on n latest samples
[buf_1, buf_1_SE] = GetTPxLatestSampleSimple(...
    bufferData(:,16:17),... % right eye, left eye: 18:19
    average_across_n_samples); % we may aggregate across n samples
t_buf_pc = lastRetrieval(end, 15); % stimulus host time of latest sample
t_buf = lastRetrieval(end, 1); % datapixx time of latest sample
```

Code: Online Data Retrieval (2)

```
% free sample-wise online data  
% retrieval without buffer  
% (eyelink style)
```

```
[pos_right, pos_left, ~, ~, ~, ~,  
t_pos_pc, t_pos] = ...
```

```
% set to 1 to get datapixx time  
GetTPxEyePosition(0);
```



But hey, online buffer-based retrieval of
10 – 16 samples takes ~0.8 millisecs

Code: Stop and shutdown

```
[rec_end_time_pc, rec_end_time] = StopTPxRecording;  
WaitSecs(0.5);  
  
% Retrieve final samples from buffer, which must be empty at the end  
[sample_i, lastRetrieval] = ReadTPxDataSimple(sample_i);  
bufferData = [bufferData; lastRetrieval];  
  
% remap Datapixx time to stimulus pc time (precise method)  
bufferData = DatapixxToGetSecs(bufferData);  
  
% shutdown TrackPixx, turn off illuminator  
UninitializeTPx;
```

Conclusion

- The TrackPixx3 is a powerful eye tracker, especially great for online applications
- TrackPixx Toolbox makes its usage easy
- Richard will increasingly use the TrackPixx now and is happy to teach and assist you if you plan to do the same
- *<https://github.com/rolfslab/TrackPixxToolbox>*