# Doing smooth pursuit paradigms in Windows 7

## Challenges and Limitations

Wilms, Inge Linda

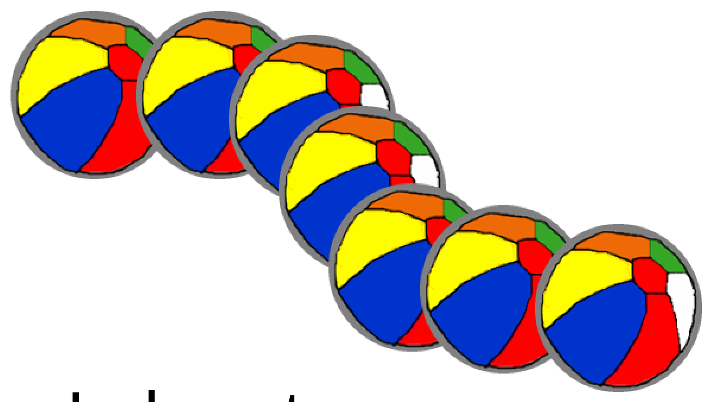
*Publication date:*
2017

**University of Copenhagen**
*Faculty of Social Sciences*
Department of Psychology

The European Conference on Eye Movements
20th August to 24th August 2017 in Wuppertal

# Doing Smooth Pursuit Paradigms in Windows 7

## -Challenges and Limitations-



**Inge L. Wilms, Ph.D.,**
Associate Professor and Leader of Brain Rehabilitation and Technology Laboratory
Contact: inge.wilms@psy.ku.dk


### Abstract
Smooth pursuit eye movements are interesting to study as they reflect the subject's ability to predict movement of external targets, keep focus and move the eyes appropriately. The process of smooth pursuit requires collaboration between several systems in the brain and the resulting action may predict strengths or deficits in perception and attention. However, smooth pursuit movements have been difficult to study and very little normative data is available for smooth pursuit performance in children and adults. This poster describes the challenges in setting up a smooth pursuit paradigm in Windows 7 with live capturing of eye movements using a Tobii TX300 eye tracker. In particular, the poster describes the challenges and limitations created by the hardware and the software.


### Background
In the clinical world, smooth pursuit eye movements are usually assessed using a so-called Wolff wand (figure 1). The clinician will ask the subject to keep his or her head still and focus on the golden top of the wand while the clinician moves the wand. The movement of the wand differs depending on the assessment but common practise is that the wand is moved in a wide circular pattern or a horizontal or vertical pattern (1). This is a rather imprecise way of assessing smooth pursuit ability for two reasons: 1) it requires that the clinician is able to move the wand in a speed and manner within the limits of the brain's ability to predict the movement and 2) it is almost impossible for the clinician to reconstruct the same movements repeatedly. This prevents the assessment of smooth pursuit before and after an intervention and confounds the reliability of the assessment of intervention induced changes in smooth pursuit ability.

Since we wanted to study the development of smooth pursuit eye movements in children, we therefore needed to create a computer-based assessment tool to reliably track and record smooth pursuit eye movements. However, we soon realized that none of the commercial providers of eye tracking equipment offer animated smooth pursuit paradigms which would allow you to control the displayed movement. The only available solutions required you to use video material of moving targets. Since we needed to know exactly when a movement started and ended to ascertain smoothness and responsiveness, we had to build our own animation software and use the SDK to communicate with the eye tracker.

**Figure 1 A Wolff Wand, a tool commonly used clinically to assess eye movements**

**Figure 2 A Wolff Wand examination to assess smooth pursuit eye movements**

### Equipment used
The eye tracker used in the study was the Tobii TX300 which tracks both eyes with a sampling rate of 300hz. The screen unit is a 23 inch TFT monitor with a resolution of 1980x1080 pixels. Vertical sync freq. is stated as being 49-75 ms with a typical response time of 5 ms (4).The PC was equipped with Intel i7-6700K 4 x 4.0 GHz processor with 8MB cache, 32 GB DDR4-2666 RAM and a GeForce GTX970 4GB Graphics card.

### Challenges in software
The animation program for the study was written by the author in C# in order to utilize the SDK interface to the eye tracker provided by TOBII (2). There were two major requirements. We needed to log every eye tracking event and we needed to insert our own task related events using a synchronized timestamp that would later allow us to match eye movements with the activity on the monitor. This was needed to measure how precisely the subject were able to follow the movement of the target and how fast they reacted to changes in movements.
We therefore captured and queued the eye tracking data passed every 3rd millisecond and inserted our own event markers into the stream of events using a synchronized timestamp to ensure we could establish the correct order later on. The writing of the queue to a data set was done asynchronously. The smooth pursuit paradigm consisted of three different tasks where the main objective was to follow the movement of a small beach ball: 1) moving horizontally back and forth at three different speeds, 2) moving in a circle at three different speeds and 3) moving in a small or large rectangular shape (Figure 4a, 4b).
When a software program displays an item on a PC monitor there is a slight delay between when the command to display an item is issued until the item actually appears on the screen. The monitor is only able to display new items 60 times per second. This may sound fast, but if you want to make the illusion of movement, you actually need to be able to display the item in a new position every 60th of a second. The animation was done by writing and deleting the image of the ball on the monitor at a very fast rate. By trial and error we found out that Windows does not supply a fast enough timer function. However, a free Multimedia Timer (3) class is available and it allows for timer interruptions to occur as low as 1 ms. With the Timer, it was possible to get the illusion of smooth movement and even control the speed from slow to medium and fast.
Figure 3 shows the flow of command when displaying and capturing the movement of the beach ball.

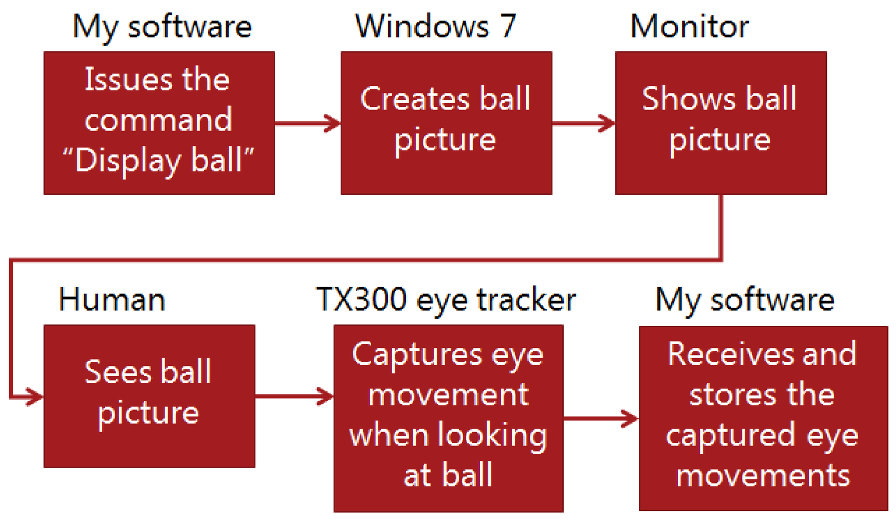



**Figure 3: The flow of command when displaying data**

### Animation and eye Tracking
The eye tracker records the position of the eyes 300 times per second. The item on the monitor only moves 60 times per second and the movement itself is restricted to be no less than the size of one pixel on the screen. This reduces the resolution of the movement and in the subsequent analysis you have to consider that the eye tracker captures data with a higher resolution than the movement itself. When comparing the precision of the smooth pursuit performance to the displayed movement, you will have to take into account that the position of the moving object is actually not moving for three consecutive events from the eye tracker or reduce the sampling rate.

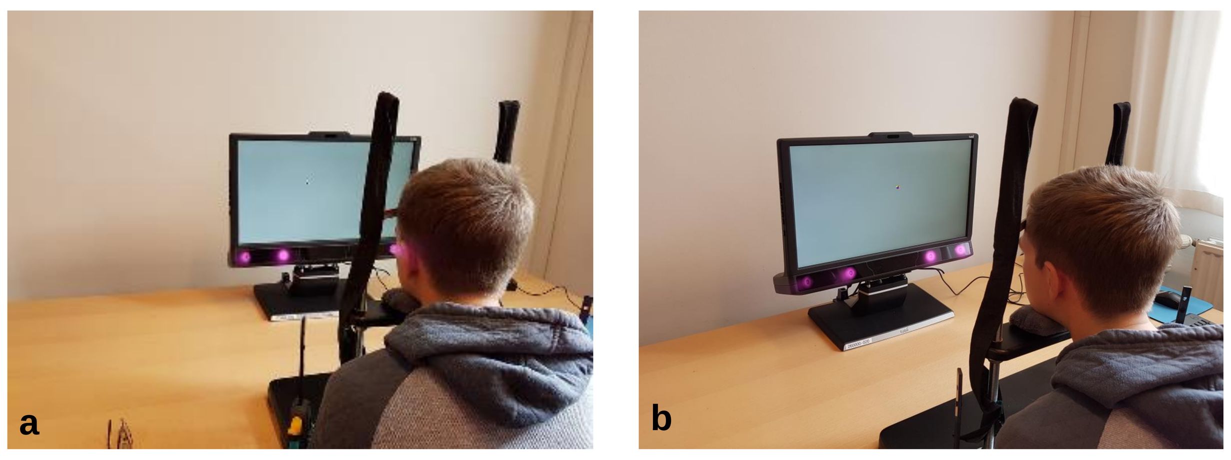



**Figure 4: The smooth pursuit paradigm. The ball is moving from left (a) to right (b).**


### References
(1) Scheiman, M., & Wick, B. (2008). *Clinical management of binocular vision: heterophoric, accommodative, and eye movement disorders*: Lippincott Williams & Wilkins.
(2) TOBII Analytics SDK Developer's guide Release 3.0
(3) The Multimedia timer for the .NET framework (URL: https://www.codeproject.com/Articles/5501/The-Multimedia-Timer-for-the-NET-Framework)
(4) TOBII TX300 Eye Tracker User Manual Revision 2