IMPLEMENTATION OF A DRIVER AWARE VEHICLE USING MULTIMODAL INFORMATION

Suwan Park
Research Engineer, HumanLAB, Daegu Gyeongbuk Institute of Sci. & Tech.
Daegu TP Venture Plant 1-202, 75 Gongdanbuk2gil, Dalseo-gu, Daegu, South Korea
+82-53-430-8449, +82-53-430-8599, psuwan@dgist.ac.kr
Joonwoo Son
Project Leader, HumanLAB, Daegu Gyeongbuk Institute of Sci. & Tech.
Daegu TP Venture Plant 1-202, 75 Gongdanbuk2gil, Dalseo-gu, Daegu, South Korea
+82-53-430-8453, +82-53-430-8599, json@dgist.ac.kr

ABSTRACT

There has been recent interest in an integrated approach to driver safety which focuses on the overlapping and interacting area of the role of driver, vehicle and road environment in driving safety. Many active safety systems such as adaptive cruise control, parking assistance and lane keeping system have been developed to target these intersecting regions. However, dynamic driver state was not properly taken into account in the safety systems because the selection of dominant attributes and modeling architectures for state detection are not fully established yet. In order to categorize driver state in terms of driver wellness, researchers in MIT suggested a modified inverted-U shaped curve which depicts the relationship between arousal level and driving performance.

This paper demonstrates an implementation of a driver aware vehicle platform to detect driver state based on the MIT wellness concept. In order to detect driver state, various overt and covert measures such as driving performance, visual attention, physiological arousal and traffic situation should be collected and interpreted. The main focus of this paper is to provide implementation techniques for the synchronized data collection and integration of inputs from multiple domains.

INTRODUCTION

It is well known that fatigue, inattention, distraction and stress strongly influence driving performance and contribute to traffic accidents and that safe driving necessitates properly managed and sustained attention (1, 2). At present, a number of advanced driver assistant systems are available, offering services to avoid hazardous situations or to facilitate driving.

As the use of in-vehicle technologies became more popular, there is concern about a concomitant increase in driver distraction arising from their use. In order to minimize the negative effects they may have on road safety, the dynamic driver state must be taken into account by these systems. However, most researches on traffic safety have been addressed through individual improvements to the car by manufacturers; improvements to the driver through education and enforcement; and improvements to the infrastructure by government. Some recent researches emphasized an integrated research to driver safety which focuses on the overlapping and interacting area of the role of driver, vehicle and road environment in driving safety as shown in Figure 1 (1, 3). In order to detect driver state, various overt and covert measures such as driving performance, visual attention, physiological arousal and traffic situation should be collected and interpreted in driving context. In the SmartCar project a vehicle embedded system was developed to evaluate driver's stress through pattern recognition methods (4). This system recorded physiological and video signals but contextual information was collected separately using questionnaires. Recently Ramon et al. (5) developed an integrated platform which analyzes driver's state through physiological measurements associated with data from other sources on the same time scale.

However, Ramon's system didn't collected some useful data to detect driver states such as visual attention measures using gaze tracker and advanced driving variables including lane position. And other data were unnecessarily missed because synchronization time was adjusted by video capture sample rate which is the lowest of the modules.

In this paper, we suggest a driver aware vehicle platform which collects a broad set of measures from multiple domains including driver, vehicle and road environment in the same time period using an elaborate timer. To achieve this function, the system implementation focuses on a soft synchronization mechanism and multimodal information fusion on PC-based Analysis System.

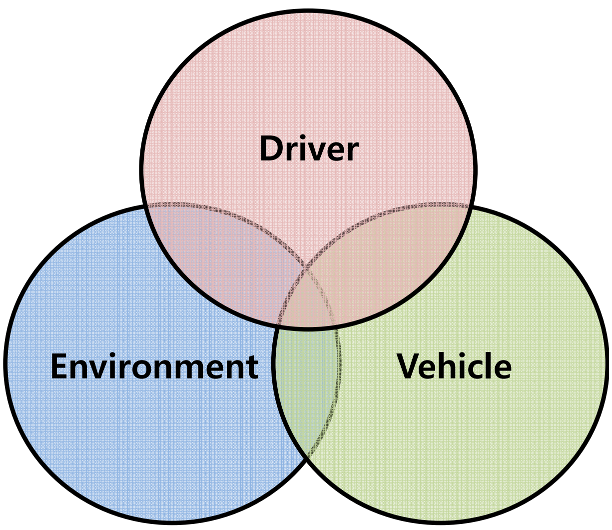

Figure 1. The primary contributing factors of driving safety

DRIVER STATE DETECTION FOR CONTEXT AWARENESS

Driver state refers to overall physical and functional characteristics indicative of features such as distraction, fatigue, attentional capacity, and mental workload. Driver state can be estimated from various overt and covert measures of driver, vehicle and road environment. The data on a driver's physiological arousal, apparent emotion, allocation of visual attention, driving style, and driving behavior should be combined with data on vehicle control and environment conditions to provide individualized and context aware measures of driver state. One specific data can't explain driver state. For example, the causes of slowing down in vehicle speed may vary. It can be due to curve road, uphill road, downhill road, obstacle avoidance, speed cameras, or combination of multiple causes. Therefore, we need to interpret driver state from a variety of perspectives. We need to look vehicle data including steering wheel angle, lateral acceleration, gas pedal and brake pedal and so on, and road geometry such as road curvature, slope from a digital map of specific location which can be found by GPS data and traffic environment such as surrounding vehicles from road view videos should be considered as road environment data. The most important data we should consider to estimate driver state is driver data which include gaze data and physiological data. For example, if someone departs his lane without signaling, we can interpret the reasons in various ways. It can be due to drowsiness, inattention, avoiding obstacles, and so on.

Therefore, an integrated platform is required for developing a driver safety system and estimating driver state. When we implement an integrated platform, the time synchronization of each data is very important to interpret the driving context using the multimodal information.

If the platform has a master timer and records key values from various modules with the same master time of short and precise interval, it may be provide with more accurate context awareness through synchronized data at any given time. The detailed soft synchronization mechanism will be described in the next section.

A DRIVER AWARE VEHICLE

SYSTEM OVERVIEW

The driver aware vehicle platform consists of six video cameras (two for driver and four for road environment monitoring), high speed and low speed CAN logger, driver gaze tracking system, global position system(GPS), image processing system for detecting lane position and angle, and physiological signal sensing system. It runs through four PCs which are Main Control PC, Sub Control PC, Driver Gaze Tracking PC, and Physiological Signal PC as shown in Figure 2. Each PC has signal acquisition module from sensor systems and real-time visualization module. They are independent, work separately to give specific information about driver's behavior, and store synchronizing data with master time.

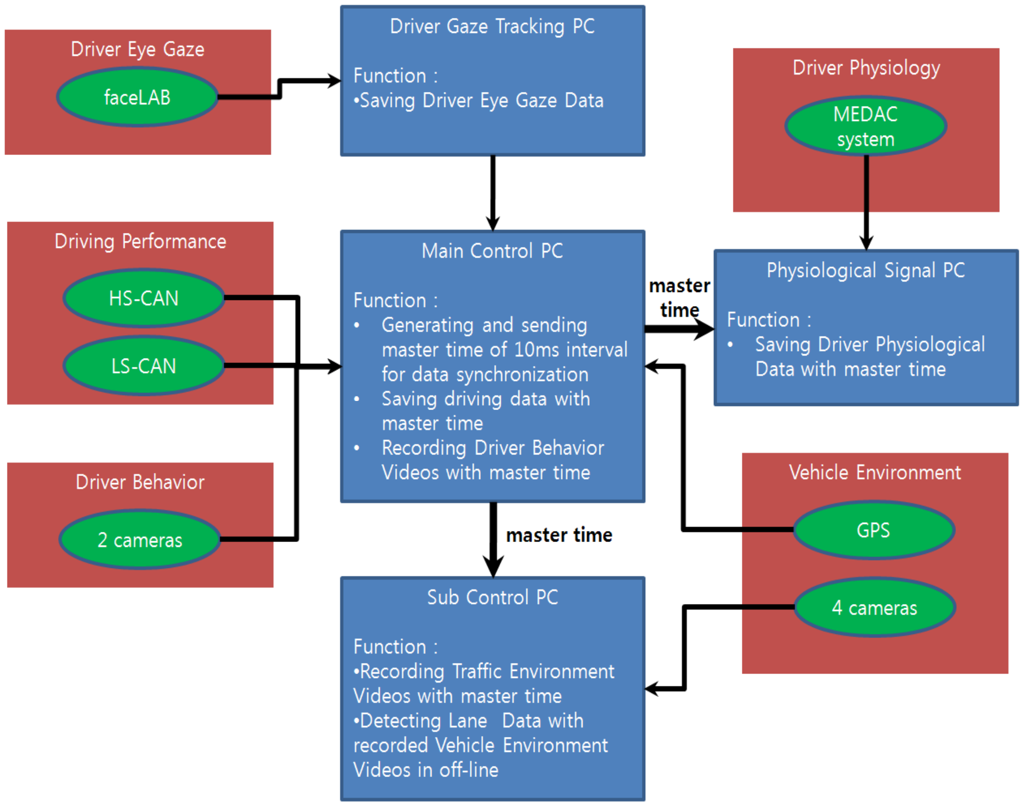

Figure 2. Concept of the driver aware vehicle system

SOFT SYNCRONIZATION MECHANISM

In order to synchronize all data, Main Control PC generates 10ms master time and broadcasts via UDP and serial port. Generally under Microsoft windows operating system, the precision of providing timer is low even though accurate multimedia timer is used. Microsoft tests have determined that lowering the interval at which the system time is updated to 2ms (as was done when the multimedia timer was introduced) has a negligible effect on processor usage. However, they report that the overall system performance is greatly reduced when a resolution of 1ms is used (6). It was the reason for choosing the clock update interval for the multimedia timer at 2ms. So the thread induced timer should be made for high resolution timer with QueryPerformanceCounter function (7, 8). We make and use master timer class in the driver aware vehicle as shown in Figure 3. More detail algorithm will be discussed in master time generating module description.

MAIN COTROL PC

Main Control PC includes 4 modules: master time generating module, CAN module, driver behavior recording module, and GPS module. They write their data with master time sent by master time generating module for data synchronization when recording their data from devices connected to themselves.

The master time generating module carries out two tasks; the one is to save the key values of other modules data on the same time scale, 10ms, and the other is to provide the master time to other modules as TimeGenerating function shown in Figure 3. The former is necessarily to read the driving context by storing the key values of other modules data. For example, the master time generating module records the frame number of videos sent from driver behavior recording module and the frame number of videos sent from Driver Gaze Tracking PC in a file on the same time scale. So at a particular time we can search correlation and information of other data and read the driving context. Some values, which are master time, CAN data

```
class MasterTimer
{
   LARGE_INTEGER start, stop, timeElapsed;
     //initialize timer
     void startTimer(){ QueryPerformanceCounter(&start); };
     //get elapsed time
     double getElapsedTime()
     {
        QueryPerformanceCounter(&stop);
        timeElapsed.QuadPart = stop.QuadPart - start.QuadPart;
        double elapsedS = (double)timeElapsed.QuadPart * m_frequency*m_unit;
        return elapsedS;
     };
     //generate and send master time to modules, and save key data sent from modules
     // on the same TIME_INTERVAL scale
     void TimeGenerating()
     {
          while (TimeThreadContinue)
        {
           nowTime=getElapsedTime();
           if (nowTime>=step-THRESHOLD)
           {
                 Send nowTime;
                 Save key data sent from modules;
                 step+=TIME_INTERVAL;
           }
      };
}
```

Figure 3. Pseudo-code of Master Timer Class

(RPM, speed, and turn signal), faceLAB data (frame number and eye gaze position), and GPS data (latitude, longitude, and altitude), are shown in the interface of Main Control PC as shown in the left picture of Figure 4. The latter enables modules to write their own data with master time as the master time generating module sends master time via UDP and serial port on the same time scale. The CAN module receives and saves objective CAN data from the vehicle (speed or command use) and information about driver's activity (e.g. steering wheel angle or pedal depression) with a time of mater timer. It obtains the signals from low speed CAN through LAWICEL CAN232 and high speed CAN through KVASER Leaf Light USB/CAN interface. The driver behavior module records two videos from the anterior and the top mounted analogue cameras as shown in the left side of Figure 4 for providing data related to the driver's activities. Video flux (NTSC) is sent to a frame grabber to perform the A/D conversion and to record the master time as overlays at the same time over the video frames. Video data are compressed to a Huffyuv format and stored in avi file-format. GPS module reads and stores GPS data (e.g. latitude, longitude, etc.) through GPS-640 device of ASCEN Company that plugs into PC USB Port.

SUB CONTROL PC

Sub Control PC performs two modules: vehicle environment recording module and lane detecting module. The vehicle environment recording module records driving road through four analogue cameras in four locations, which are at the front, the rear, the left, and the right side, as shown in the right picture of Figure 4 for providing data related to the driving context and traffic situations. It sends frame numbers of 4 video frames to Main Control PC via TCP/IP on every master timer tick. Video flux (NTSC) is processed in the same way as driver behavior module. In off-line, the lane detecting module computes lane position and angle of a vehicle from the front-view and right-side rear view videos.

Figure 4. Screen shots of Main Control PC (left) and Sub Control PC (right) during a driving experiment

DRIVER GAZE TRACKING PC

The faceLAB^{TM}, the head and gaze analysis tool, is used for driver gaze tracking in Driver Gaze Tracking PC (9). The faceLAB is a software package that uses a set of stereo cameras as a passive measuring device. Images from the cameras are analyzed to work out characteristics of a subject's face, including the current position and orientation in 3D space, the gaze direction and several other measurements. It saves driver gaze tracking data and sends eye gaze position and frame numbers of cameras, which give driver gaze tracking data, to Main Control PC via UDP.

PHYSIOLOGICAL SIGNAL PC

Three physiological measures are recorded through MEDAC System/3 in physiological signal PC: Electrocardiogram (ECG), skin conductance, and respiration wave. Electrocardiogram (ECG) monitors the electrical activity of the heart over time for tracking heart rate and variability. Skin conductance and temperature sensor monitor electrodermal activity and skin temperature change from a finger.

DATA ANALYSIS

DATA ANALYSIS TOOL

Our specific software is used for data analysis as shown in Figure 5. This tool shows the flows of synchronized data at once according to change in the master time. The flows of all data in certain interval of the master time can be seen and the corresponding video frame can be displayed. Values of the current time cursor (Red line rectangle and circle) refers to synchronized data from current time. By positioning the current time cursor (Red line rectangle) on data, the corresponding video frame and other data can be displayed. These functionalities are very useful and convenient for data analysis related to driving and context. Other functions such as zooming, scaling and simple operation make data exploration and visualization easier.

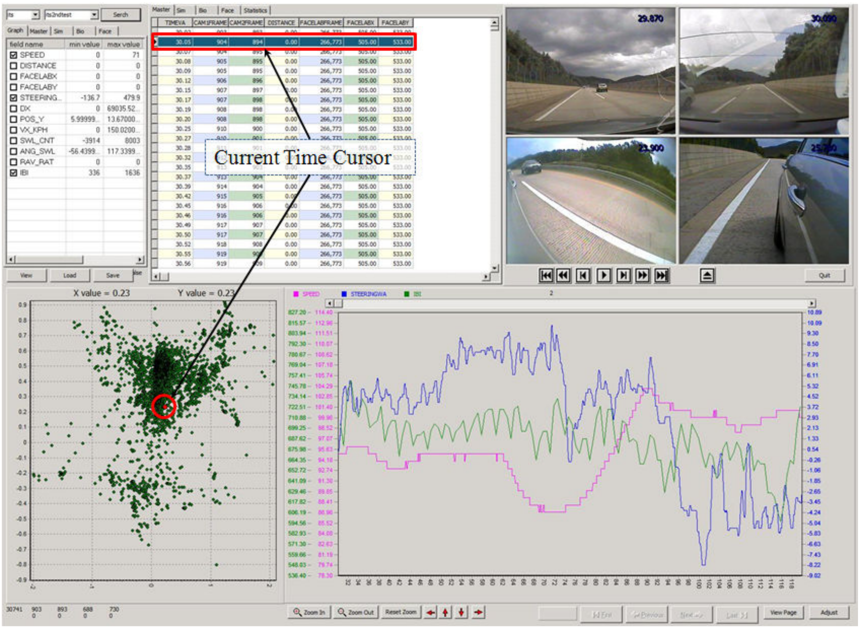

Figure 5. Concurrent data analysis tool

EXPERIMENT DATA

We collected data through our driver aware vehicle in the route chosen for the road experiments as shown in Figure 6. A participant drove 31 km from the location B to A. He was told not to exceed speed limit (100 km/h) while driving.

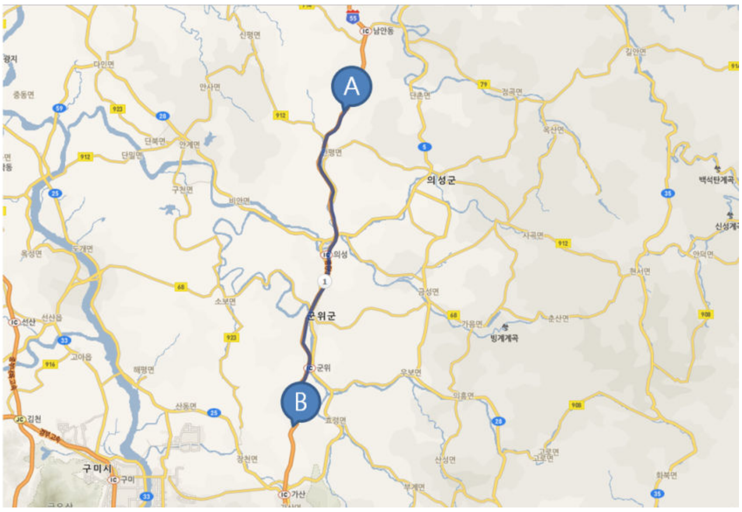

Figure 6. Route for the road experiments

As shown in Figure 7, Speed (pink line), steering wheel angle (blue line), and interbeat interval (green line) are described in our analysis tool. It shows that some events are occurred to the participant by observing overall changing of decreased speed and IBI and increased fluctuation of steering wheel angle after the time of red dashed line. We could look into the reasons by searching other synchronized data such as GPS data, driver behavior videos, gaze tracking data, and vehicle environment videos. Actually he started to perform a secondary task, which requires cognitive resources, in the time of red dashed line in Figures 7.

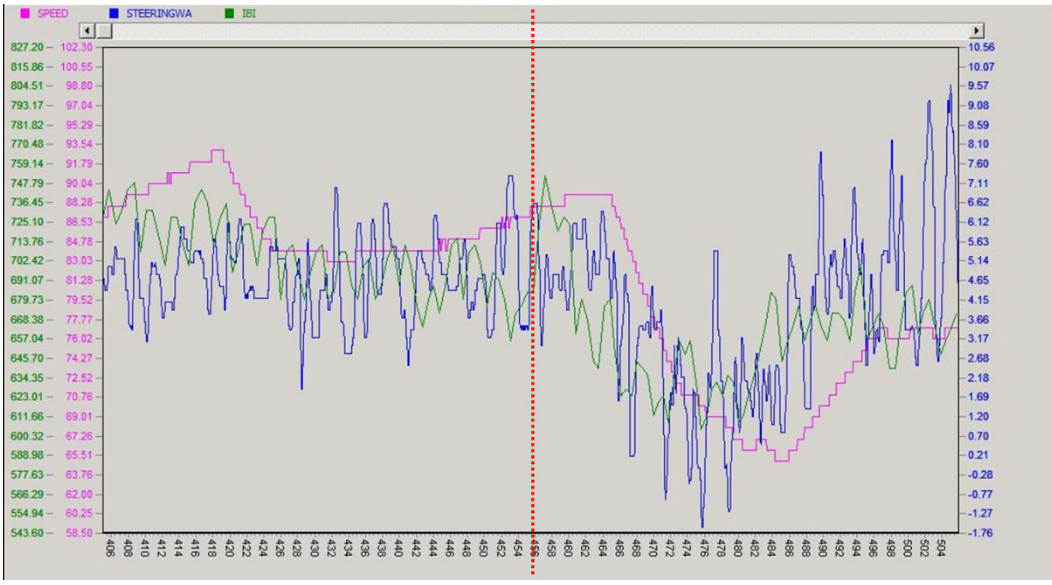

Figure 7. An example of driving behavior changes by adding a secondary task

An illustration of the gaze trail across for the participant is shown in Figure 8. It shows the participant experienced visual tunneling in right picture relative to left picture. We could look into the reasons by searching other synchronized data such as driver behavior videos and physiological data. Actually, left picture is gaze trail during driving only session and right picture is gaze trail under dual task condition (driving and secondary task).

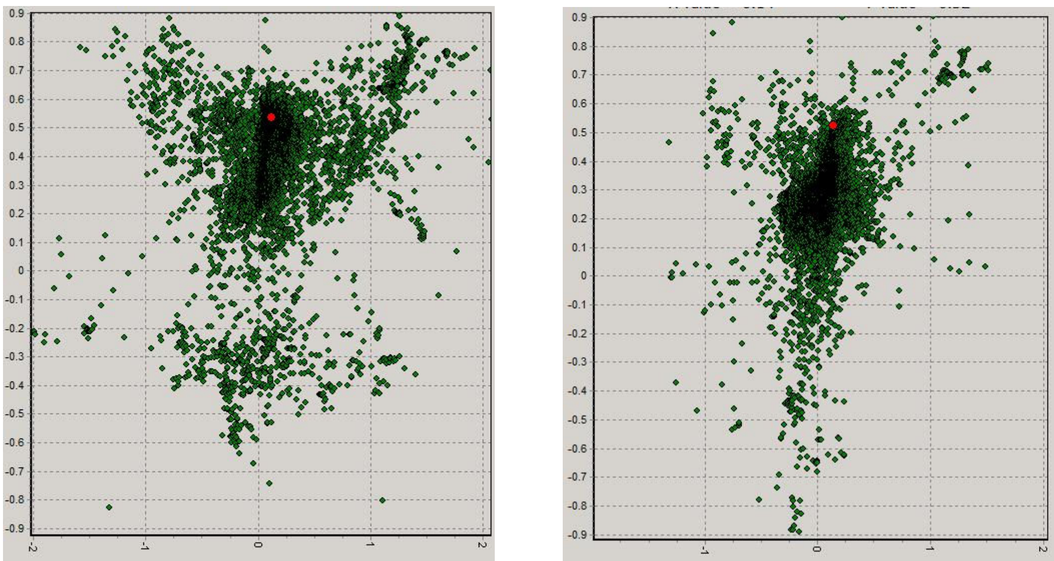

Figure 8. Gaze trail of driving only task (left) and dual task (right)

CONCLUSION

We implemented a driver aware vehicle platform to detect driver state using various information from a driver, a vehicle and an environment. This platform records multimodal information through six video cameras, high speed and low speed CAN logger, driver gaze tracking system, global position system (GPS), and physiological signal, and provides the synchronized data collection and integration of inputs from multiple domains with a master timer. Thus driver state and contextual information can be evaluated through different indicators. This multivariate analysis is aimed at improving reliability. We made and used data analysis tool that contextual events can easily be found through time chronology. So, we could search what happened at a particular time by recorded contextual data in field test. The future goal is to add real time lane detecting system and to develop a driver distraction detecting algorithm through our driver aware vehicle platform and analysis tool.

ACKNOWLEDGEMENT

This research was supported in part by Daegu Gyeongbuk Institute of Science and Technology (DGIST) Research Program of the Ministry of Education, Science, and Technology (MEST), Implementation of More Efficient Transportation System Program(06 C01) of the Ministry of Land, Transport and Maritime Affairs (MLTM), and Industrial Strategic Technology Program of the Ministry of Knowledge Economy (MKE).

REFERENCES

(1) Reimer, B., Coughlin, J. F., and Mehler, B., “Development of a Driver Aware Vehicle for Monitoring, Managing & Motivating Older Operator Behavior”, in *Proceedings of the ITS-America* (Washington, DC, 2009).

(2) Son, J., Reimer, B., Mehler, B., and et al., “Age and cross-cultural comparison of drivers’ cognitive workload and performance in simulated urban driving”, *International Journal of Automotive Technology*, Korean Society of Automotive Engineering, Seoul, August 2010, pp. 533-539.

(3) Brooks, C.A., Rakotonirainy, A. and Maire, F.D., “Reducing Driver Distraction through Software”, in *Australasian Road Safety Research Policing Education Conference* (Wellington, 2005)

(4) Healey, J.A., Picard R.W. SmartCar: Detecting Driver Stress. ICPR, p. 4218, *15th International Conference on Pattern Recognition* (ICPR'00, 2000).

(5) Ramon, C., Clarion, A., Gehin, C., Petit, C., Collet, C., & Dittmar, A. An integrated platform to assess drivers's physilogical and functional states. *Proceedings of the 30th Annual International IEEE EMBS Conference* (Vancouver, 2008).

(6) Peng, J.T. WINDOWS HARDWARE AND DRIVERS CENTER. Guidelines For

Providing Multimedia Timer Support (2002).
(7) Johan, N., Implement a Continuously Updating, High-Resolution Time Provider for Windows, MSDN magazine, 2004.
(8) Janno, G. and Derrick, K. "Design of a High Resolution Soft Real-Time Timer under a Win32 Operating", in *Proceedings of SAICSIT 2005*, pp. 226 – 235
(9) FaceLAB4 User Manual, Seeing Machines, Canberra, Australia (2008).