Experimenting Kinect interactions in the car

Mario Chiesa
Istituto Superiore Mario Boella
via Boggio 61
10138 Torino
chiesa@ismb.it

Alessandro Branciforti
Istituto Superiore Mario Boella
via Boggio 61
10138 Torino
branciforti@ismb.it

ABSTRACT
As an input device for pervasive computing, the Kinect perfectly fits into several scenarios and use cases. However, pervasive computing typically crafts services and experiences in a new fashion, splitting them into micro-interactions, that needs special attention for designing the best Kinect's usability. We are exploring these new requirements into the car environment, with some insights that can be extended to other scenarios. Our experiments, after initial in-lab tests, are conceived within a new concept for a user interface and visualization for assistive driving.

Categories and Subject Descriptors
H.5.2 User Interfaces – Input devices and strategies

Keywords
Interaction design; Kinect; Natural user interfaces

1. INTRODUCTION
Kinect was born originally as a motion sensing device for video games. Recently, Microsoft released a PC-version of the device, aiming to open the implementation of the interface into several other domains, potentially everywhere, and hopefully generating what Microsoft calls "The Kinect Effect" [1]. Kinect in games and in many other scenarios can be used in different ways. Typically the use cases and the context of interactions require users to interact with the system for a time that can last from few minutes to hours. The concepts of pervasive computing and pervasive interactions open up new scenarios where each interaction with the system, and therefore with the Kinect sensor, can last only few seconds. This rises up new requirements and needs, which we have tried to explore in a specific setting.

1.1 The pervasiveness of touchpoints
In fact, pervasive scenarios redefine the human-computer interaction model and distribute services' touchpoints throughout the whole people everyday experience: through spaces, along time and mindsets.

AutomotiveUI'12, October 17-19, Portsmouth, NH, USA.
Adjunct Proceedings

For this reason, instead of having 'sessions' where users interact for an amount of time with a service, people have and will have more and more a series of single interactions with multiple services. It could be a simple status update about the tracking of an order, the selection of an option from a short list of alternatives, the request for a stock exchange index, the confirmation for a specific action (to sell stocks, buy a ticket, book a flight), etc.

1.2 Touchpoints for micro-interactions
In other terms, people's life will be more and more interrupted with and disseminated of several micro-interactions or micro-transactions. These micro-transactions define all together the lifecycle of the service, which has to be designed and implemented for being 'used' and interacted through several platforms, in different contexts and mindsets. This has an immediate effect on the dynamics of each single interaction. They have to be fast, reliable, and simple. In one word: natural. So from this point of view, user interfaces like the Kinect fit well into these scenarios; however, having to be fast, reliable and simple, they introduce also some additional requirements, that have to be fulfilled.

2. PERVASIVE INTERACTIONS
2.1 A three-steps process
To highlight the main focus of our paper, we introduce a simple model to describe the process through which an interaction with a service happens. In brief, every interaction (in a pervasive scenario, but this is true everywhere) can be roughly described as structured into three steps:

- An *initial step*, where the user is recognized by the system: it can be a login page, an identification system, or a sort of 'wake-up' for the touchpoint.
- A *core step*, where the user, once the contact is successfully established, interacts with the system or the service: it can be short and simple, like an on/off switch or the selection from an options' list, or complex and longer like the management of the entire service.
- A *final step*, where the user can logout, or simply moves away from the touchpoint; this step can sometimes be an optional one: the user simply stops to interact with the system, without logging out or going away.

2.2 The three-steps process for Kinect
When a user interacts with the Kinect, this is how typically the three steps are perceived by her/him:

- An *initial step*, where the user lets her/himself to be recognized by the system: 'waving' her/his hand in front of the camera, or 'raising up' a hand for the same reason, or 'standing up' in front of the sensor: this 'wakes up' the Kinect and allows the system to identify her/himself

- A *core step*, where the user moves her/his body or arm or hand in front of the sensor, and performs some activities and tasks within the application
- A *final step*, where the system 'understands' by itself when the interaction has terminated.

2.3 The three-steps in pervasive scenarios

Usually the core step takes much more time than the other two, and for this reason the main focus is on the user interaction during this phase, while the user performs several tasks and actions with the service, as explained before.

In a pervasive scenario the situation is different: the time spent to pass through the *initial step* and the time needed in the *core step* to do a single micro-interaction are similar. For this reason, it changes the way we have to conceive the whole process.

3. KINECT INTERACTIONS IN THE CAR

In our activities, we have focused on automotive scenarios and on the car environment. In the car of the future drivers will be able to interact not only with the car itself, its devices, components, systems and sub-systems, but also with the outside: location-based services, social network-based services and infotainment services.

3.1 Main requirements

Drivers should be able to instantly receive information about traffic updates, weather forecasts, etc. and to ask the car for them very easily, without distractions. For us, each of these tasks is a micro-interaction with the car system, considered as a whole. Think about this example:

1. First, the driver wants to check if the traffic in the downtown city is ok.
2. A minute later, the driver wants to reply to a received message.
3. Later, the driver raises the volume of the car stereo.
4. After another five minutes, the driver tells the car to find the closest parking slot available.

Each micro-interaction will have its *initial*, *core* and *final* step and each shall be **fast**, **natural**, **and reliable**. In this paper we focus our attention on the 'fast' and 'reliable' requirements, as attributes that can affect also the 'natural' aspects of the interaction.

To have fast micro-interaction, one thing we have to reduce is the time needed to complete the initial step, or the step needed by the car to understand that the user wants to interact with it. We do not consider login, or identification processes, but simply the 'wake up' phase for the Kinect. Therefore, in this initial phase of the project, our main focus is how to reduce this 'wake up' time.

In our experimental settings, we consider the whole windscreen as a surface where the system will project its output, visually. The output consists of simple information like the ones described above, that need to be pointed and selected, with ease, and - again fast. Systems already available in the market, as aftermarket products or shipped from the factories, typically project the information on smaller head-up displays (HUDs).

3.2 Experimenting with the code

Starting from the examples available in the open source literature, we have explored how to tweak and refine timing, algorithms and strategies to reduce the time needed for the initial step.

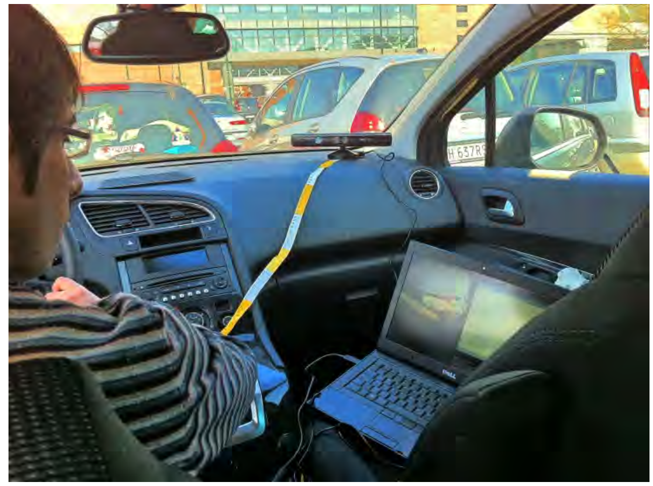

Figure 1 - Reference configuration for a Kinect in a car

Our approach is, after an initial "wave" gesture, to always track the movements; this task was quite obvious and simple to do as the hand of the driver is, at least, close to the steering wheel for the whole duration of the journey. If the passenger next to the driver cause some interference by obscuring the camera or the IR sensor for a while, normally the Kinect sensor is still able to maintain the identification of the driver's hand and therefore to visualize the hand marker. To tweak core interactions to be reliable and fast, we tried several methods. We present here three of them.

3.2.1 The "Wait over marker" method

Our first approach for a system capable of drawing some information on the car's windshield and tracking the user's hand movements over them is a "wait over marker" method.

We draw a single point over the screen every time the position of the tracked hand changes. This will result displaying a "tail" that follows the hand movements. The coordinates for the tail were given by simple indexes capturing the x and y axis values from an available Kinect method. A second circle is draw with a fixed delay time (this can be a user's choice). When the hand stops for a short period in a specific activation area, it triggers the associated action. This method is easy to understand and master, due to its affinity to the 'classic' point-and-click systems. On the other hand, the user/driver has to share her/his attention while driving the vehicle to follow the cursor on the windshield.

3.2.2 The "Push to Activate" method

Three activation areas are displayed on the screen (in green in Figure 2). The Kinect sensor reads the relative position of the x axis with a triggering sequence of the three different activation areas, just moving the tracked hand left or right. This method doesn't require any cursor and at least one of the activation areas is always targeted. It just mimics the "push" gesture (thus changing the z axis) and the target will fire / end the event. This method, however, shorten the distance between the sensor and the tracked hand while the driver is 'pushing': due to the car environment, with an already limited distance, this can decrease the precision of the tracking.

Figure 2 – Activation areas for the "Push to activate" method

3.2.3 The "Circle to Activate" method

Another interesting way to introduce Kinect interactions in automotive environment is to detect circular movements using available methods. We have introduced some changes: we shaped three different activator rings. Rather than change the radius, as in the open source code, activating one ring equals to fire an event. The program checks if the position of the hand marker is placed within an activator ring (in blue in Figure 3); then, if a circular gesture is captured, the system draws a second ring (in yellow, in Figure 3); this ring changes its radius until it matches the activator ring. When the two circles collide, the event is fired.

Figure 3 – Detection of a circular movement

Another thing to note is that the gesture (a circle drawn in the air) recognition can be associated to the radius of the activation's circles; bigger rings equal to wide circular hand movements, while small rings equal to narrow hand movements, until to track just the wrist rotation.

4. THE PILOT STUDY

Our project is aimed at the creation of a prototype able to use new interface paradigms for the human-machine interaction in the automotive sector. One of the purposes of this project is the creation of a configurable prototype to act as a demonstrator able to use new interface paradigms.

4.1 The setup

In order to explore the possibility of an in-car gesture detection system, based on Kinect interactions, we have designed and prototyped a system with two distinct applications, both written in Processing. One application generates data such as speed, acceleration and steering angle, reading and elaborating data from a Logitech G27 steering wheel paired with its pedals and clutch, and writes them to another application that generates the visual output for the windshield. Another application receives data generated by the first one, reads data from other user interface components, together with the Kinect sensor, and creates the visualization for the windshield display.

4.2 The concept for the user interface

The interaction with the system can happen using the Kinect or using more 'traditional' in-car device elements. For the purpose and the scope of this paper we focus our attention only on the interaction with the Kinect. The system can provide two kinds of visualization, projected directly on the inner surface of the whole windshield (at least in the concept, while in the initial proof-of-concepts we projected the images on a smaller surface, similar to a HUD). This type of visualization is somehow close to the Pioneer/Microvision product concept [2]. The first visualization is related with the current speed of the car and the corresponding stopping distance. This information is visualized as a sort of 'shadow', as it was projected by the car in front of itself, with the apparent length equal to the stopping distance.

The second type of information is a set of widgets available to the drivers, to choose and allow to be displayed aside to the 'shadow' (visible in green in Figure 4). These widgets can show data or visualizations about speed, gas mileage, driving info, etc. Widgets can be selected while the car is not moving. During driving, the driver can only choose between selected widgets as in a carousel: only simple interactions are performed during driving, while more complex settings require full attention from the driver, so they are available during stops. The only widget that can't be moved is the "safe brake distance slider"(visible in green in Figure 5).

When the car moves, the system removes the circular control and the driver can see the activated widgets such as the "safe brake distance slider"; this takes place in the lower center area of the driver's point of view and highlights the safe brake distance according to the current speed of the car. In this case the driver can actually see how far (close) the car ahead is from his distance slider in order to adjust (reduce) his speed; the peculiar function of this widget makes its position fixed in front of the driver point of view.

4.3 The preliminary results

We have tested the different interaction techniques described above within our test environment. As preliminary results of the test pilot, one of the suitable solutions involving Kinect sensors is to display only three widgets areas at a time to maintain the cognitive load at an adequate level during driving.

Figure 4 - Selection between widgets

Figure 5 - The interface during the tests

The corresponding activation areas are located at the top of the sensor's action range; those locations avoid the risk to activate something while maneuvering the steering wheel.

The method chosen to deal with the activation zones is a mix between the "wait over marker" and the "push to activate" mentioned above. The latter is used without the "push" gesture: after one second waiting over the marker, the widget is activated.

5. CONCLUSION

The concept of overlaying visual feedbacks on the windshield, with proper luminosity and position, allows a better usage of important information without distracting the driver's eyes out from the line of sight with other visualization devices such as smartphone or on-board monitor, so the user can remain focused on the road. On the other hand, sharing space on the windshield means we have to apply some limitations in terms of quantity (max 3 widgets at once) and design; the widgets, intended to add information and not to hide part of the road, are made as simple as possible with light color and with a transparency degrees chosen by the user in the setup menu.

The usage of a Kinect in an automotive setting seemed to be a very promising novel approach to reduce driver distraction while interacting with the automotive UI, but sometimes we found that drivers need to look at visual feedbacks for a longer time, to search for the cursor, or other feedback, on the windshield.

This can be dangerous in a driving scenario and this leads us to choose the "activation zone" as described before, somehow limiting the usage of Kinect. Our aim is to go deeper into this investigation and move from current results and methods to new and original ones.

6. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support to this work by the regional project "AUTUMN - AUTomotive hUMan Net", a collaborative research project partially funded by Regione Piemonte and FinPiemonte through the instrument of the "Polo di Innovazione ICT".

7. REFERENCES

[1] The Kinect effect – Xbox.com. http://www.xbox.com/en-US/Kinect/Kinect-Effect

[2] Pioneer Cyber Navi http://pioneer.jp/press-e/2012/pdf/0511-1.pdf