VENTURI
(FP7-288238)
immersiVe ENhancement of User-world Interactions

D6.3 Use case(s) implemented on VENTURI integrated platform V1

Due date of deliverable: 30-09-2012
Actual submission date: [01-10-2012]

Start date of project: 01-10-2011
Duration: 36 months
## Summary of the document

<table>
<thead>
<tr>
<th>Document Code:</th>
<th>D6.3 Use case(s) implemented on VENTURI integrated platform V1 – v1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last modification:</td>
<td>30/09/2012</td>
</tr>
<tr>
<td>State:</td>
<td>Ready for submission</td>
</tr>
<tr>
<td>Participant Partner(s):</td>
<td>ST-Italy, Metaio, Fraunhofer, FBK, INRIA, STE, e-Diam</td>
</tr>
</tbody>
</table>
| Editor & Authors | Editor: David Siorpaes, Paul Chippendale  
Authors: e.g. Javier Campos (e-Diam), David Siorpaes (ST-Italy), Olivier Pothier (STE), Daniel Buhrig (HHI), Selim Ben Himane (metaio) |
| Fragment: | No |
| Audience: | ☑️ internal  
☐ public  
☐ restricted |
| Abstract: | This document is VENTURI prototype deliverable D6.3 “Use case(s) implemented on VENTURI integrated platform V1” and illustrates the year-1 VENTURI demonstrator. |
| Keywords: | Use cases implementation, demonstrator, prototype, integrated platform, requirements |
| References: | Refer to the corresponding section at the end of the deliverable |
# 1. Table of Contents

2. Executive Summary .......................................................................................................................... 5  
   Audience .......................................................................................................................................... 5  
   Summary .......................................................................................................................................... 5  
   Structure .......................................................................................................................................... 5  
3. Introduction ..................................................................................................................................... 5  
4. Augmented reality gaming demo application ................................................................................... 5  
   Game organization .......................................................................................................................... 5  
   Game description .............................................................................................................................. 6  
   Mission 1: Extinguish the fire at the building (single player) ............................................................. 6  
   Mission 1: Extinguish the fire at the building (multi player) ............................................................... 6  
   Mission 2: Burger King" mission (single player) .............................................................................. 6  
   Mission 2: “Burger King” mission (multi player) ............................................................................. 7  
   Observer mode .................................................................................................................................. 7  
   Game control ..................................................................................................................................... 7  
5. Demonstrator development ............................................................................................................ 8  
6. VeDi-1 demonstration installation and operation ............................................................................. 10  
   Demo installation ............................................................................................................................. 10  
   Demo operation: tracking ................................................................................................................ 10  
   Tracking process creation ............................................................................................................... 12  
   Demo operation: gameplay .............................................................................................................. 13  
7. Key Features and implemented requirements .................................................................................. 17  
   Occlusion models ............................................................................................................................. 19  
8. Conclusions ..................................................................................................................................... 20  
9. References ....................................................................................................................................... 20
2. Executive Summary

**Audience**
This deliverable is confidential, restricted for members of the VENTURI consortium only.

**Summary**
This document describes the use case that has been implemented on the VENTURI integrated platform V1. The VENTURI Year one Demonstrator consists of an Augmented Reality game on which gameplay occurs on a miniaturized city model.

This document provides detailed instructions on how the prototype has been implemented on the VENTURI enabled device (VeDi), as well as operative instructions on how to replicate the prototype setup.

**Structure**
This deliverable is structured as follows: Section 3 briefly introduces the key features of the Year 1 VENTURI prototype. Section 4 details the nature of the prototype and provides a high level view of its possible different implementations that have been discussed during the progress of Work Package 6. Section 5 illustrates how the development of the prototype evolved in the final demo application, highlighting both its key features and the practical issues that have been encountered during the process. Section 6 summarizes the requirements elicited from the use case definition phase that are actually used and tested by the Year 1 VENTURI prototype. Section 7 concludes the document providing detailed instructions on how to download, install and operate the prototype.

3. Introduction
The VENTURI V1 demonstrator, from now on referred as “VeDi-1” prototype, consists of an Augmented Reality game showing advanced Augmented Reality capabilities on top of the ST-Ericsson U8500 platform. In particular, the demonstrator highlights the links to the hardware and software architecture requirements:

- reported in Deliverable D2.1.1 “Use cases, application definition and system requirements for STE U8500-based Platform”,
- specified in deliverable D2.2.1 “Early Detailed Design Specifications for STE U8500-based Platform”
- implemented in deliverable D2.4.2 “First STE U8500-based Platform baseline delivery, implementing D2.2.1 specifications”

4. Augmented reality gaming demo application
The VENTURI year one prototype consists of a standalone single/multi-player game based on 3D markerless tracking running on the first version of VeDi device. The game takes place indoors on a table-top 3D game board. The game is built on top of the metaio Mobile SDK which is the Mobile Creator AR framework.

**Game organization**
The organization of the game consists of two distinct phases: a scene tracking part and a game one. During the tracking phase the scene of the gameplay, which is a 1:200 scale city model, is analyzed and recorded. This preliminary phase is just performed once. The second part is the actual game, where the user interacts with Augmented Reality objects with characters acting and moving within in the real city model.
Game description

Two main gaming scenarios with different objectives have been identified during the development of Work Package 6. They can both be played either in single player mode or multi-player mode. This section provides a high level description of the candidate scenarios that have been identified for VeDi-1 game demonstrators, which are reported here for reference purposes.

The Actual implementation of the final game is described in more details in “Demo installation and operation” chapter.

Mission 1: Extinguish the fire at the building (single player)

- The device offers the user the options to choose between SP or MP environment. (This case is for SP).
- The user can choose between different missions, always with a time limit.
- The user selects the mission called "Fire mission" or "Burger King" mission.
- The system presents the instructions that the user will follow to control the objects in the game, so as the steps he/she has to follow to accomplish the goal.
- The user will move to the Fire Station, so focusing with the device will see the virtual fire truck. Clicking on it, the user will have the control of the truck.
- Rotating the device, the user will turn the vehicle and clicking on the on screen buttons with arrows, it will allow him/her to move the virtual truck thru the city, which will avoid collisions of the vehicle, so as the occlusion to get the effect when you cannot see the track on your scope.
- In the mission, the user will drive the truck till the fire's place, so with the on screen button for this function, it will allow the user to shoot water to the fire to extinguish it.
- Other possibility: How about if the user can rescue people from some buildings using the truck's stairs?
- Game will end if the user has got the goal (saving people, extinguishing the fire, etc.) or the time is up.

Mission 1: Extinguish the fire at the building (multi player)

- User 1 (U1) begins the game with his/her VeDi and choose MP game. So U1 has to wait for the User 2 (U2) to join the game. It will be done thru an interface in which the device will offer a list of user actually connected to join who you want to play with.
- Connection between devices will be thru server or BT or any other way to connect them.
- U1 and U2 have the option to be the "savior" or the "damager", so they'll have to compete in the game, in which the "damager" will try to avoid the success of the "savior" user.
- Users will stand around the physical table. The "bad" user begins turning around to choose where he/she wants to put a fire. The bad user goes with the device and point the place that is going to be burning. User will use on screen buttons to fire to the building (for instance).
- The good guy will have to play as in the SP game, going to the fire place to extinguish the fire with the on screen controls.
- The bad guy won't be allowed to fire the truck, so the game will have an end :)
- The bad guy will have the possibility to burn 3 places, so that the good guy will have enough time to extinguish every fire place.
- Game will end if the good guy extinguishes the 3 focus of fire or if the time is up.

Mission 2: Burger King" mission (single player)

- The user will enter the game, and the system will ask him/her to go to the Burger King site to choose the vehicle that will be used to accomplish the "Burger King mission". He/She will select one by focusing on it and then clicking on the screen.
- From this moment, the user will receive on-screen warnings to know what and where he/she has to take to the customer. Warnings will appear on screen at random times until a maximum of three, as the user sees the time running in the corner of the screen.
- These warnings will be minimized on a side on the screen to be seen whenever the user clicks on them. Such messages or warnings may appear at any moment, so the user might be making a service whilst the message appears on the screen.
- The user will have to pick up the goods in the Burger King and then will have to drive till the customer’s place, and give it to the customer.
- The maximum number of goods that he will be able to take is two, so the user can make two deliveries without going back to the Burger King.
- The game will end when all of the services are done or if the time is up.

Mission 2: “Burger King” mission (multi player)
- User 1 (U1) begins the game with his/her VeDi and chooses MP game. So U1 has to wait for User 2 (U2) to join the game. It will be done through an interface in which the device will offer a list of users actually connected to choose whom you want to play with.
- User will choose to be “the good guy” or “the bad guy”, dictating who has to do particular tasks.
- The goal of the bad guy is to stop the good guy from accomplishing his goal of making the deliveries. The bad guy will be able to set up to 10 obstacles to delay the delivery time of the good guy.
- The bad guy will have an interface in which he has 10 on-screen thumbnails of the objects that can be left in the way (e.g. stones, working signs, etc.) so he/she will be able to click on them and focus where in the city he/she wants to leave them.
- As objects are left in the city, the good guy will have to change his way to avoid them or use some kind of tool to take them out of the way, losing time in that action, so it will delay the delivery.
- The bad guy can place the objects as time goes by, not necessarily placing them all at the beginning of the game.
- The game will end when all of the services are done or if the time is up.

Observer mode
During the VENTURI face to face meeting in Valencia, it was proposed to add an interesting option to allow a third participant to act as “Gameplay observer” for demonstration purposes. In this mode, the gameplay scenes are actually rendered on a third device, which does not participate actively to the gameplay but just observes in real-time how the gameplay is actually developing.

Game control
This section specifies the user interface that the game player is exposed to during the gameplay and defines how the user interacts with Augmented Reality objects in the screenplay.

- The user will experiment during the game, controlling objects through the device by at pointing them and selecting them by clicking on the virtual objects on the screen.
- As the user rotates the device left or right, the selected object will turn in the same direction.
- To move the virtual objects, the user will have on screen buttons with up/down arrows to move it and other controls according to the situation (fire, etc.)
- The user controls the vehicle from the rear of the truck (as shown in Figure 1).
5. Demonstrator development

Game application has been developed on top of metaio’s Mobile Creator Software Development Kit and the Android Operating system APIs, as shown in Figure 2.

![Diagram showing the architecture of the demonstrator](image)

**Figure 2 - VeDi-1 demonstrator architecture**
The game application has been coded using Eclipse 3.7.2, using the metaio Mobile Creator Software Development Kit version 3.1.1 and running on the U8500 platform based on Android 4.0.1.

Initial game development efforts have concentrated on the Single Player Burger King gameplay, and the Firefighter and Multiplayer games have been given lower priority. In fact, some technical difficulties have been encountered during the practical development of the VeDi-1 demonstrator. Since the VENTURI technologies embodied in both the two gameplays described above is exactly the same, and multiplayer game is not bringing significant added value to the demonstration itself but “nice to have” accessory features, the VENTURI consortium decided to concentrate efforts on ensuring that the gameplay itself is smooth, convincing and, overall, enjoyable.

Most critical issues have been investigated and successfully fixed. Others have been worked around for later analysis and feedback for next generation VeDi devices. The issues that have been encountered and successfully overcome or worked around are reported in Table 1.

**Table 1 – Demonstrator development issues**

<table>
<thead>
<tr>
<th>Issue ‘brief description’</th>
<th>Solution/workaround and impact on the development cycle</th>
</tr>
</thead>
</table>
| When recording for more than two minutes the device hangs. | Workaround: Reset board  
Impact: longer development cycles |
| Sending 3D tracking data via email is not working in the 30% of cases. | Workaround: accessing the SD card to retrieve manually the 3D tracking log files.  
Impact: longer development cycles |
| Device camera gets stuck after installing application Android Package repeatedly. Camera doesn’t run (screen appears black) and even Android standard Camera application, reports “Camera has stopped”. | Workaround: Reset board  
Impact: longer development cycles |
| Memory problems: application package updates end up in “low on memory space” messages. | Workaround: manually uninstall and re-install Android application packages during development  
Impact: longer development cycles |
| Touchscreen is not responding. | Workaround: re-adjust touchscreen connector and reset board.  
Impact: longer development cycles |
| Occasionally, after disconnecting the device while charging, the device does not respond. | Workaround: Reset board  
Impact: longer development cycles |
| 3D tracking does not align correctly. When moving around the city model in the 3D tracking version, VENTURI buildings move towards the side you move, so they don’t stay in the correct place to get the occlus- | Solution: fix has been implemented by Metaio |
### 6. VeDi-1 demonstration installation and operation

This section illustrates how to install and operate VeDi-1 prototype. A U8500 board and an account on VENTURI Steerforge are needed (see VENTURI deliverables D2.4.1 and D2.4.2).

**Demo installation**

Demonstration is based on the VENTURI platform release labeled “STE platform deliveries DV1.3” that can be downloaded from the VENTURI Steerforge site at the following address:


You must follow the installation instructions provided in deliverable D2.4.2 “First STE U8500-based Platform baseline delivery, implementing D2.2.1 specifications” for software platform installation guide.

Once the appropriate Software platform has been flashed on U8500 board, download game demonstrator installer package from the following link:

https://www.steerforge.com/file/download.php/197/127/p56_r80/VeDi1-demo.apk

And install it using Android Debug Bridge tool from a DOS command line as follows:

adb install VeDi1-demo.apk

ADB installation procedure, as well as all information related on how to operate the U8500 board, is reported in deliverable D2.4.2 “First STE U8500-based Platform baseline delivery, implementing D2.2.1 specifications”.

**Demo operation: tracking**

The Creator Mobile [1] app enables the creation of tracking configurations for the VENTURI Y1 demonstrator on the VeDi device without the need of programming. The game organizer positions a set of markers in the neighborhood of the 3D game board (a miniature city model) and the app starts to learn the appearance and geometry of the 3D environment thanks to the user moving the mobile device around the board. The markers are used to define the origin and the orientation of the coordinate system (see Figure 3). They also help to provide reliable camera pose estimation during the 3D tracking configuration creation and the scale of the environment. In case more than one marker is used, online inter-marker calibration is performed.
The markers also help during the authoring process using the Creator since the point clouds are not an intuitive representation of the environment. A special version of the official Creator Mobile has been created adapted to the VeDi device constraints of Y1. This version takes into account the VeDi device specific changes to the 180 degree camera rotation issue (reported in previous deliverables) and handles the gravity sensor data differently. More information on the tracking process and the Mobile Creator tool can be found in deliverable D2.3.1 “First implementation of junaio-based AR framework for STE U8500-based Platform”.

Tracking recording is made with 20cm markers from metaio (number 80, 81 and 82) with normal lighting conditions and using the Creator Mobile application. Markers are available at the following link:

https://www.steerforge.com/file/download.php/197/126/p56_r80/MarkersTracking3D.pdf

Figure 4, Figure 5 and Figure 6 show how the markers are arranged around the city model during the 3D tracking phase.
Tracking process creation
To create the 3D tracking process, we have to follow these steps:

- We have to place the marker(s) to record the 3D environment. In order to correctly understand the real-world scale of the model, we have to place more than one marker the scene, in most cases it is possible to use one of these as a reference point.
• We use the CM software from metaio on the VEDI device (the ideal situation is to record and test the 3D tracking from the same device) and use the ‘start’ option to record a sequence for at least 30 seconds to create an adequate 3D representation of the environment. We must move around the model slowly and take care to maintain the coloured points visible and the 3D axis over the marker in order to get a better result.

• We send the 3D tracking file (it will be on the SD card) by mail from the app. We un-compress the zip file and use it in the ‘assets’ folder of the Android project.

**Demo operation: gameplay**

For correct operation, the city model should be arranged in the same fashion as the acquisition stage, i.e. as shown in Figure 7

![City Model Arrangement](image)

**Figure 7 - City model arrangement**

The Burger King game is structured as follows:

• The user has two minutes to play the game.

• As the game starts, the user has to click on the burger that will always appear over the BK, being closer that a distance of ‘200’ units, shown in the left part of the screen.

• As the user takes the burger, he/she has 30 seconds to deliver it to the “red” point that he/she will have to find in the city.

• Once the place to deliver has been found, the user needs to drive using the on screen controls to a distance closer than ‘200’ units, as shown on the screen, and then click on the “red” flashing point to deliver the order.

• As the goods are delivered, another burger will appear over the BK and the user will have to go there and again be closer than ‘200’ units or he/she will not be able to collect the goods.

• The goal is to deliver a maximum number of orders within the game time.
The game is started by the user tapping on the “VeDi demonstrator” icon on Android OS. Once the application is launched and initialized, the user is presented with a screenshot like the one depicted in Figure 8. Note that the “VENTURI” solid blocks superimposed on the city model represent the ‘preliminary’ occlusion models and are represented only for game debugging/evaluation purposes.

![Figure 8 - Game Launch Screenshot](image)

As the game is developed on top of the city model it is possible to use occlusion reasoning to obtain greater 3D realism, the button “Ocl” can be used to activate/deactivate the occlusion models. The game is started by tapping on the “Start” button on the upper right part of the screen. Once the game is started, a hamburger icon appears on the Burger King building and on the left part of the screen the following information is presented (Figure 9):

- Time to deliver: Timer counts down from 30 seconds.
- Burger: Distance to the burger. It has to be less than 200 units.
- Delivery point: Distance to the place to deliver the order. It has to be less than 200 units.
- Delivered: Number of successfully delivered orders.
- Failed: Number of fails, as timer expires.
Clicking on the burger initiates the delivery process. Burger will disappear and the user will have to look for the delivery point, go with the car until it is at a distance closer than 200 units and then click on the delivery point to confirm the delivery of the order. The user will drive the car using the on screen controls to go forward, backwards, left and right (Figure 10).

Delivery points will be represented by a red flashing sphere that will appear randomly over the city model. Using the occlusion models enables the virtual 3D models to be visualised over the actual buildings (Figure 11).
Figure 10 - Car closeup
7. Key Features and implemented requirements

This section reports which of the requirements defined in D2.1.1, specified in D2.2.1 and implemented in D2.4.2, have been integrated and exploited in the prototype. It also highlights other key features that have been integrated in the demonstrator which were not part of the requirements list.

Table 2 is an excerpt of D2.2.1, limited to the HW and SW functional requirements, with three additional columns, that indicate:

- whether the requirement has been implemented in the Platform delivery DV1.3, on which the current VeDi-1 demonstrator is built,
- whether the requirement is used by the VeDi-1 demonstrator,
- whether the requirement is validated by the VeDi-1 demonstrator.

<table>
<thead>
<tr>
<th>Req. ID</th>
<th>Analysis Status</th>
<th>Fulfilment in DV1.0</th>
<th>Effort</th>
<th>Risk</th>
<th>Implemented in DV1.3</th>
<th>Used in VeDi 1.0</th>
<th>Validated by VeDi 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW Functional Requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HF1</td>
<td>Done</td>
<td>Yes</td>
<td>None</td>
<td>None</td>
<td>Already implemented</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HF2</td>
<td>Done</td>
<td>Partial</td>
<td>None</td>
<td>None</td>
<td>Partial</td>
<td>No (1)</td>
<td>No</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>---------</td>
<td>------</td>
<td>------</td>
<td>---------</td>
<td>--------</td>
<td>----</td>
</tr>
<tr>
<td>HF3</td>
<td>Done</td>
<td>Yes</td>
<td>None</td>
<td>None</td>
<td>Already in DV1.0</td>
<td>Yes (2)</td>
<td>Yes</td>
</tr>
<tr>
<td>HF4</td>
<td>Done</td>
<td>Yes</td>
<td>None</td>
<td>None</td>
<td>Already in DV1.0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HF5</td>
<td>Done</td>
<td>Yes</td>
<td>None</td>
<td>None</td>
<td>Already in DV1.0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HF6</td>
<td>Done</td>
<td>Yes</td>
<td>None</td>
<td>None</td>
<td>Already in DV1.0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HF7</td>
<td>Done</td>
<td>Yes</td>
<td>None</td>
<td>None</td>
<td>Already in DV1.0</td>
<td>No (3)</td>
<td>No</td>
</tr>
<tr>
<td>HF8</td>
<td>Done</td>
<td>a posteriori check</td>
<td>None</td>
<td>No indication</td>
<td>Nothing to be done</td>
<td>Yes</td>
<td>Not yet (4)</td>
</tr>
<tr>
<td>HF9</td>
<td>Done</td>
<td>Yes</td>
<td>None</td>
<td>None</td>
<td>Already in DV1.0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HF10</td>
<td>Done</td>
<td>Yes</td>
<td>None</td>
<td>None</td>
<td>Already in DV1.0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HF11</td>
<td>Done</td>
<td>Yes</td>
<td>None</td>
<td>None</td>
<td>Already in DV1.0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HF12</td>
<td>Done</td>
<td>Yes</td>
<td>None</td>
<td>None</td>
<td>Already in DV1.0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**SW Functional Requirements**

| SF1 | Done | Yes at Platform level | None | None | Already in DV1.0 | Yes (5) | Yes |
| SF2 | Done | Yes (limitation)     | None | None | Nothing to do   | Yes | Yes |
| SF3 | Done | Partial              | None | None | Nothing to do   | Yes | Yes |
| SF4-1 | Done | Yes             | None | None | Nothing to do   | Yes (6) | Yes |
| SF4-3 | Done | No               | Medium | Medium | Yes | No (7) | No |
| SF4-4 | Done | No | Medium | Medium | Yes | No (8) | No |
| SF5 | Not Analyzed | a posteriori check | None | No indication | Nothing to be done | Yes | Yes |
| SF6 | Not Analyzed | a posteriori check | None | No indication | Nothing to be done | Yes | Yes |
| SF7 | Yes | a posteriori check | None | No indication | Nothing to be done | Yes | Not yet (9) |
| SF8 | Yes | Yes | Medium | Medium | Yes | No (9) | No |
| SF9 | Yes | Partial | Medium | High | No | No | No |
| SF10 | Not Analyzed | No indication | No indication | No indication | Nothing to be done | No (10) | No |
| SF11 | Not Analyzed | No indication | No indication | No indication | No | No | No |
(1) Connectivity is not used in Single Player mode (Vedi-1). This requirement will be validated in Multi Player mode (Vedi-1.1)

(2) Raw Sensors (direct outputs from Sensor HW) are used in Vedi-1 for aiding the feature tracking. Added value has been validated.

(3) Vedi-1 does not implement any audio feedback.

(4) This requirement (1 hour continuous operation) has not been tested yet. It will be done during the first public exhibition of Vedi-1, at the InsideAR event.

(5) Vedi-1 demo has been successfully tested on devices with different screen resolutions; although the prototype itself comes with fixed screen resolution.

(6) This requirement has been validated by both the Vedi-1 and the dataset recording application, part of DV4.1. However, only the low level API of the sensor data is working properly. The sensor fusion API (part of the ICS release) is not functional on Vedi-1, due to several issues needing further investigation. Strictly speaking, SF4-1, as described in D2.2.1, only mentions the low-level data.

(7) This requirement has not yet been tested at demonstrator level. Unitary tests have been conducted at the end of the T2.4.1 phase, but the requirement still needs to be validated from the user perspective. Its individual validation would be difficult and should be envisioned jointly with SF4-4 and SF8.

(8) This requirement (same time base for video and sensor data) could not be tested yet in Ved1-1, since the feature tracking needs to use the new SurfaceTexture API in order to get the Video frame timestamps.

(9) This requirement (timestamping of the video frame in an accurate manner) could not yet be tested, since the feature tracking needs to use the new SurfaceTexture API in order to get the Video frame timestamps.

(10) This requirement has been classified as optional in D2.4, and has not been implemented on the Vedi-1 prototype since the Monkeyrunner Android tool [2] already fulfils this requirement.

This analysis highlights that some of the requirements that were elicited, analyzed and implemented in DV1.3 are not yet validated by the Vedi-1 platform. The next release of Vedi-1.1, due shortly, will partly solve these issues, especially by enabling the multi-user gaming mode. However, validation work remains, either by an evolution of the Vedi-1.1 demonstrator, or from an evolution from the D4.1 dataset recording application, through the common and accurate timestamping of video and sensor data (requirements SF4-3, SF8 and SF4-4).

Occlusion models

To enable a realistic augmentation of the virtual car into the real-world playing field, occlusions caused by physical objects need to be handled explicitly: whenever any of the (real) miniature city houses in the game impedes the line of sight from the user to his (virtual) car, only those visible parts of the vehicle should be rendered.

As this requires an accurate 3D model of the real scene, Fraunhofer HHI has reconstructed all of the miniature houses used in the game. In contrast to sparse feature-based reconstruction approaches, such as the one used for visual 3D tracking, the 3D models delivered for the first year prototype exhibit closed surfaces, making them suitable for use as occlusion models.

To create the 3D models, all of the houses were placed on a turntable and photographed under various viewpoints and observation angles, creating a very dense multi-view dataset. The images were segmented and fed into a shape-from-silhouette reconstruction algorithm, producing 3D models at very high resolution (several million triangles per object). To allow for efficient rendering on the mobile device, simplified versions of those models with 5-15 thousand triangles per object were computed.

Three example reconstructions are presented in Figure 12; note that the models shown here are un-textured, since texture is not needed to implement the occlusion handling.
8. Conclusions
The VENTURI year one prototype represents the final outcome of several steps accomplished throughout the VENTURI development process: use cases definition, requirements inference, platform specification and finally platform integration. The Year one prototype has a twofold function: on one side, it concludes the first development cycle highlighting the Augmented Reality platform requirements that have been conceived and integrated during the first year of the project; on the other side, it will serve as the first reference platform for benchmarking, optimization and new requirements definition activities, targeting VENTURI prototypes for the year two and year three timeframes.

9. References