

**SIXTH FRAMEWORK PROGRAMME
PRIORITY IST-2002-2.3.1.8
Networked Audiovisual Systems**

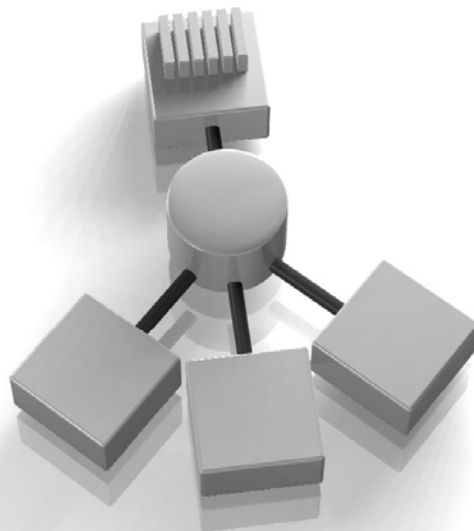


Uni-verse project

Deliverable D 4.3

Slim Rendering Client

June 1, 2006



STREP project

Project acronym: Uni-Verse

Project full title: A Distributed Interactive Audio-Visual Virtual Reality System

Proposal/Contract no.: 002228

Distribution: Public

WP 4.2 RENDERING CLIENTS – SLIM RENDERING CLIENT	3
1 Introduction	3
2 Description	3
3 Results	4
4 Conclusions	5
5 Availability	5
6 References	6

WP 4.2 Rendering clients – Slim Rendering Client

1 Introduction

The Slim Rendering client in this project is defined as a Verse renderer for PDAs and Pocket PCs. The development was focused on these devices mainly because Smartphones were only rarely available on the market at the start of the project and had (and still have) very small display sizes. Speaking of usability and ergonomic for a visualization of 3D data such a small device is not very usable.

2 Description

The slim client was implemented using Embedded C++ on PocketPC. OpenGL functionality was successfully realized on the PDA using ported versions of the emulation libraries Klimt [1] and PocketHAL [2]. PocketKnife [3] was used to provide a development environment serving standard application functionality. A STL Port [4] was used to realize standard library functionality for the renderer. The Omni-Thread library enables the usage of threads on a slim device like a PDA and was used for the implementation of the Slim Rendering Client.

The client is implemented using three threads. One thread is responsible for the network communication; another one interprets the Verse data and renders it into a bitmap. The third one steers the user interface (e.g. the input widgets) and displays the rendered bitmap. Figure 1 shows the communication structure of the Slim Rendering Client. Additionally a new widget system was developed to enable the application to display a window type and user interface parts within the rendering context. The pen and button navigation was realized to interact with the 3D scene. A Verse network connection is possible using Bluetooth and WLAN to connect to a Verse server.

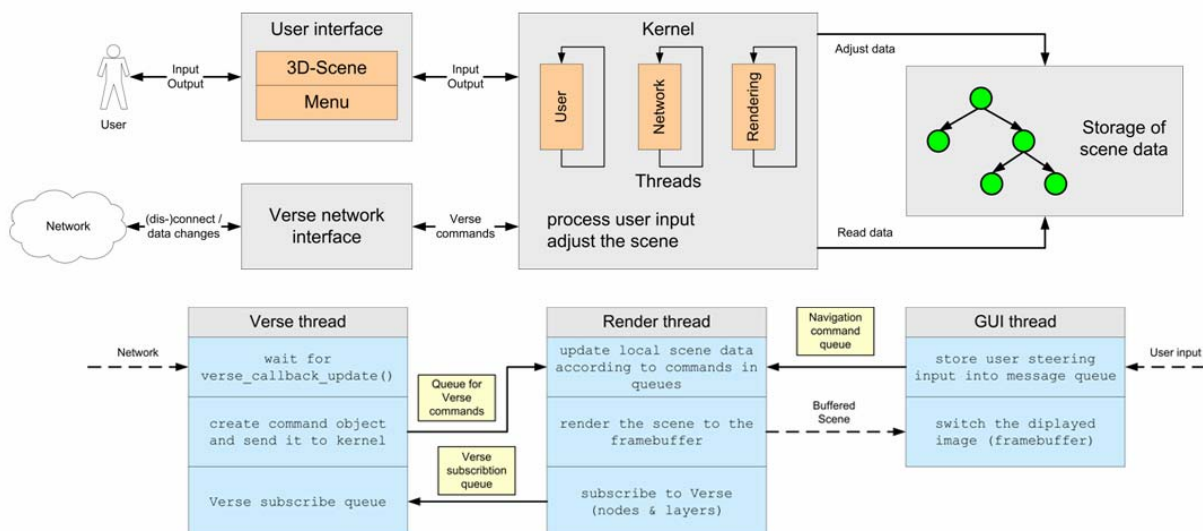


Figure 1: Communication structure of the Slim Rendering client.

The Verse data structures are simplified in the Verse thread for the usage in an embedded system. Therefore not all Verse commands are interpreted by this client. E.g. the computation of 64 Bit values doesn't make sense in this context because even floating point values must be emulated (using PocketHAL). The commands for such verse actions are therefore dropped or adapted to reasonable commands for an embedded system.

3 Results

The client was tested using Verse version R5 on an ARM, XScale on WindowsCE. The below Figures show a simple rendered object (Figure 2) the verse node tree view (Figure 3) and textured scene rendered on the desktop and on a PDA downloaded from the same Verse server (Figure 4). All models were uploaded to the Verse Server from 3ds files using a 3ds Verse pusher.



Figure 2: Reference model "Cow" rendered on the Slim Rendering client.

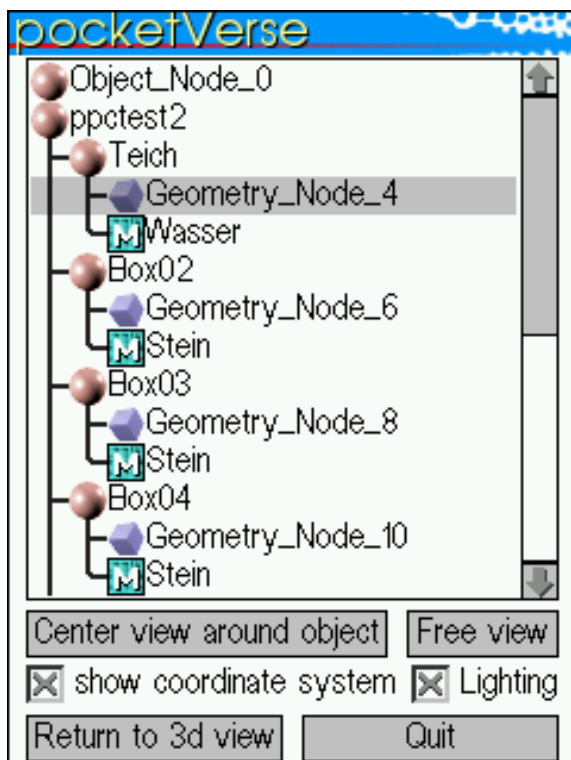


Figure 3: Verse Node Viewer



Figure 4: A textured verse scene downloaded from the Verse server and rendered on a desktop Verse viewer and at a PDA using the slim rendering client.

4 Conclusions

The Slim Rendering client enables the visualization of Verse data on slim and mobile devices like PDAs. Using a wireless LAN connection it is possible to have a mobile Verse data visualization available.

However, the amount of data to visualize is limited according to the strong hardware restrictions of such devices especially looking at processor performance or graphics acceleration. But the latest hardware developments on the market show that these constraints will more and more disappear and the performance features will become more and more equal to today's available standard desktop systems.

5 Availability

The current version of the Slim Client is already available as Open Source at <http://pocketverse.sourceforge.net>.

6 References

- [1] Klimt OpenGL emulation library: <http://www.studierstube.org/klimt>
- [2] PocketHAL: <http://pockethal.droneship.com>
- [3] PocketKnife: http://sourceforge.net/project/showfiles.php?group_id=96814&package_id=103635
- [4] STL Port: <http://www.syncdata.it/stlce/index.html>