UNIVERSITÉ LOUIS PASTEUR STRASBOURG

Thèse présentée pour obtenir le grade de

Docteur de l'Université Louis Pasteur

Strasbourg I

Discipline : Image - Photonique - EEA

par Markus Feisst

Étude de protocole et réalisation d'un système de réalité augmentée mobile

Soutenue publiquement le 21.06.2006

Membres du jury

Directeur de Thèse :	Joël Fontaine, Prof. Dr.
Codirecteur de Thèse :	Andreas Christ, Prof. Dr.
Rapporteur Interne :	Patrick Meyrueis, Prof. Dr.
Rapporteur Externe :	Antoine Delley, Prof. Dr.
Rapporteur Externe :	Jan-Wilhelm Fischer, Prof. Dr.
Examinateur de Thèse :	Bernard Keith, Prof. Dr.

Acknowledgement

First I want to thank my supervisor Professor Dr. Joel Fontaine and my co-supervisor Professor Dr. Andreas Christ for their support and their confidence in me and my work. They provided me with the technical support as well as the financial support via the European project “Rhenaphotonics” and the University of Applied Sciences Offenburg.

Furthermore, I want to thank Professor Dr. Antoine Delley, Professor Dr. Jan-Wilhelm Fischer, Professor Dr. Patrick Meyrueis and Professor Dr. Bernard Keith for taking the time to be part of my jury and their helpful comments on my work.

I want to express my gratitude to the members of the “Laboratoire de Systèmes Photoniques” for their support, their help during my work and during the preparation of the defence. Special thanks go to Dr. Ing. Thierry Blandet for his great technical support and his help with the French translation as well as to Assistant Professor Dr. Patrice Twardowski for his help with the coordination of the project groups. Thanks to all students of “Laboratoire de Systèmes Photoniques” which contributed with a project work or a Thesis to develop this work.

Last but not least I want to express my gratitude to all members of the “Mobile Communication and Guides Wave Laboratory” at the University of Applied Science Offenburg. Thanks go to all students which contributed with a Thesis in order to help me to develop my work. I want to express my thanks to the (former) members of the laboratory, notably Jelena Mitic, Diter Dos Santos for the helpful and challenging technical discussions and Dr. Gisela Hillenbrand for her help with the English translation.

Last but not least I want to thank Priscilla Owen for her help with the English translation and Oliver Möller and Frank Buri for their support and knowledge in printing.

June 2006, Markus Feisst

Table of Contents

List of abbreviations:

Abbreviation	*Full expression*	*Comment*
2D	Two dimensional	
3D	Three dimensional	
3DS	Autodesk file format	3D file format
1G	First Generation	
2G	Second Generation	
2,5G/2+G	2,5/2+ Generation	
3G	Third Generation	
AMPS	Advanced Mobile phone system	Mobile communication network
AMR	Adaptive Multi Rate	Mobile phone codec
ASCII	American Standard Code for Information Interchange	Character encoding
API	Application Programmable Interface	Software development
AuR	Authentication Centre	Component of mobile telecom. network
BSC	Base Station Controller	Component of mobile telecom. network
BTS	Base Transceiver Station	Component of mobile telecom. network
CDMA	Code Division Multiple Access	
CDMA2000	Code Division Multiple Access	
CF	Compact Flash	Storage media
CRT	Cathode Ray Tube	Monitor type
CS	Circuit Switched	
CSD	Circuit Switched Data	Mobile telecommunication data connection type
D-AMPS	Digital AMPS	Mobile telecommunication system
DCS	Digital Cellular System	Mobile telecommunication system
DXF	Drawing eXchange Format	3D description and file format
EDGE	Enhanced Data rates for GSM Evolution	Mobile telecommunication data connection type
ETSI	European Telecommunication Standards Institute	Standards organisation
FHO	University of applied Science Offenburg	
GDL	Graphics Description Language	3D description language
GGSN	Gateway GPRS Support Node	Component of mobile telecom. network
GIF	Graphics Interchange Format	Image format
GMSC	Gateway MSC	Component of mobile telecom. network
GPRS	General Packet	Mobile telecommunication data connection type
GSM	Global System for Mobile communications	Mobile telecommunication system
GZIP	Gnu ZIP	Compression format
HLR	Home Location Register	Component of mobile telecom. network
HMD	Head Mounted Device	Virtual reality device
HSCSD	High-Speed Circuit-Switched Data	Mobile telecommunication data connection type
HTML	Hyper Text	
HTTP	Hyper Text Transfer Protocol	
I/O	Input/Output	
IMT-2000	International Mobile Telecommunications-2000	Standardisation organisation
IS-95	Interim Standard 95	Mobile telecommunication standard
ISDN	Integrated Services Digital Network	Telecommunication System
ITU	International Telecommunication Union	Standards organisation
JPG/JPEG	Joint Photographic Experts Group	Image format
J2ME	Java platform Micro Edition	Software interpreter
JSR	Java Service Request	Standard

Abbreviation	***Full expression***	***Comment***
KB/s	Kilo Bytes per Second	Transmission speed unit
LCD	Liquid Crystal Display	Monitor type
LSP	Laboratoire des Systèmes Photoniques	
LZ	Lempel-Ziv	Compression algorithm
LZ77/LZ78	Lempel-Ziv and Huffman coding	Compression algorithm
LZC	Lempel-Ziv and COMPRESS	Compression algorithm
LZW	Lempel-Ziv-Welch	Compression algorithm
M3G	Mobile 3D Graphics	Standard
ME	Mobile Equipment	Component of mobile telecom. network
MMC	Multi Media Card	Storage media
MMS	Multimedia Messaging Service	Description language
MP3	MPEG-1 Audio Layer 3	Audio format
MSC	Mobile Switching Centre	Component of mobile telecom. network
NFF	WorldToolkit	3D description and file format
NMT	Nordic Mobile Telephone	Mobile telecommunication system
NSV	NullSoft Video stream	Video format
OBJ	Wavefront 3D object	3D description and file format
OGG	Operation Good Guys Open multimedia bit stream container format	Audio format
OS	Operating System	
PCS	Personal Communication System	Mobile telecommunication system
PDA	Personal Digital Assistant	
PDF	Portable Document Format	Document format
PLNM	Public Land Mobile System	
PNG	Portable Network Graphics	Image format
PS	Packet Switched	
PSTN	Public Switched Telecommunication Network	
QoS	Quality of Service	
RA	Real Audio	Codec used by RealNetworks
RMI	Remote Method Invocation	
RNC	Radio Network Controller	Component of mobile telecom. network
RTSP	Real Time Streaming Protocol	
RTP	Real-time Transport Protocol	
SD	Secure Digital	Storage media
SDL	Scene Description Language	3D description language
SGSN	Serving GPRS Support Node	Component of mobile telecom. network
SIM	Subscriber Identification Module	Component of mobile telecom. network
SMS	Short Message Service	Text exchange method in GSM
SSL	Secure Socket Layer	
STL	Stereo Lithography	3D description and file format
TCP/IP	Transmission Control Protocol/Internet Protocol	Data transmission
TDMA	Time Division Multiple Access	
TD-SCDMA	Time Division-Synchronous Code Division Multiple Access	
TFT	Thin Film Transistor	Monitor type
TLS	Transport Layer Security	
TV	TeleVision	
UMTS	Universal Mobile Telecommunication System	Mobile Telecommunication System

Abbreviation	*Full expression*	*Comment*
USIM	UMTS SIM	Component of mobile telecom. network
UTF8	8-bit Unicode Transformation Format	Character encoding
VGA	Video Graphics Array	Computer display standard
VPN	View Plane Normal	
VR	Virtual Reality	
VLR	Visitor Location Register	
VRC	Viewing Reference Coordinates	
VRML	Virtual Reality Modelling Language	Description language
VRP	View reference Point	
VUP	View Up vector	
W-CDMA	Wideband CDMA	Mobile telecommunication standard
WAE	Wireless Application Environment	
WAP	Wireless Application Protocol	
WDP	Wireless Datagram Protocol	
WLAN	Wireless Local Area Network	Network type
WTA	Wireless Telephony Agent	
WTLS	Wireless Transport Layer Security	
WTP	Wireless Transport Protocol	
WMA	Windows Media Audio	Used by Microsoft
WML	Wireless Markup Language	Description language
WMV	Windows Media Video	Used by Microsoft
X3D	Extension to VRML	Description language
XML	eXtensible Markup Language	
XVL	eXtensible Virtual world description Language	

Abstract French

La réalité virtuelle 3D offre à l'utilisateur la possibilité d'une représentation réaliste d'informations. Avec l'aide des langages de description pour la réalité virtuelle (par exemple VRML, X3D) un moyen puissant est disponible. Aujourd'hui de nombreux sites Web utilisent cette technologie pour offrir au visiteur une information tridimensionnelle. Du fait d'une bande passante toujours plus importante sur l'Internet, la complexité des scènes peut être augmentée, ce qui nécessite de la part des systèmes clients une capacité de traitement de plus en plus grande. Sur des ordinateurs personnels, le traitement des données 3D est réalisé par un système de visualisation attaché au logiciel de navigation. L'objectif de ce projet de recherche est d'offrir à des utilisateurs d'appareils mobiles ces mêmes fonctionnalités. Les appareils mobiles (téléphone portable) sont limités dans leur capacité de traitement, d'affichage et d'interface utilisateur. Bien que la mémoire dans les nouveaux appareils augmente sans cesse, en comparaison avec les ordinateurs ils ne comportent que de faibles capacités en mémoire centrale et en stockage. Le raccordement à un réseau téléphonique sans fil limité en bande passante présente des temps de latence plus élevés et n'est pas aussi fiable qu'un réseau câblé. Par conséquent les données 3D pour la réalité virtuelle doivent être optimisées avant leur transmission, c'est dans ce but que le concept de "VRML proxy" transparent a été développé.

Plusieurs langages de description existent pour décrire des scènes virtuelles 3D, parmi ces différents formats VRML (Virtual Reality Modeling Language) a été choisi. C'est un format bien connu et largement répandu pour la diffusion de données 3D sur l'Internet. Une évolution de cette norme s'appelle X3D. En ce qui concerne les fonctionnalités qui nous intéressent il n'y a aucune différence entre VRML et X3D. Néanmoins le travail a été réalisé sur la base de VRML, et donc c'est cette norme qui est mentionnée et employée ici. L'intérêt d'avoir la possibilité de visualiser et de manipuler des données 3D sur des appareils mobiles a été montré par plusieurs groupes de recherche, dans des domaines comme le divertissement, l'éducation ou l'aide instantanée.

Objectifs et concepts

Le but est de transférer puis de visualiser des données de réalité virtuelle sur les appareils mobiles et de pouvoir présenter ensuite ces informations en vue monoscopique ou stéréoscopique. Pour atteindre ce résultat il y a trois manières possibles d'agir sur le contenu original des données afin d'en optimiser le contenu pour la visualisation sur les appareils mobiles:

- côté serveur
- au niveau intermédiaire du proxy
- côté client

L'essentiel de notre travail a porté sur:

- l'étude du concept de "proxy client", côté client,
- l'étude des appareils mobiles et de leurs capacités, de la définition du "proxy server“ (plus tard appelé "proxy VRML"),
- l'étude des contraintes liées aux réseaux de télécommunication et des langages de description de scènes 3D adaptés aux mobiles (X3D, VRML).

Vue d'ensemble du système

L'illustration 1 montre la vue d'ensemble de l'architecture du système qui est déployée afficher des données 3D sur des appareils mobiles. Pour les appareils mobiles plusieurs technologies existent pour leur connexion à l'Internet. En Europe la communication est établie en utilisant le GSM, le HSCSD ou le GPRS et l'UMTS. Le système EDGE est davantage utilisé aux États-Unis comme étape intermédiaire vers l'UMTS.

Du fait que le GSM et HSCSD utilisent une transmission à commutation de circuit, une transformation des données sous forme de paquets doit être faite pour accéder à l'Internet.

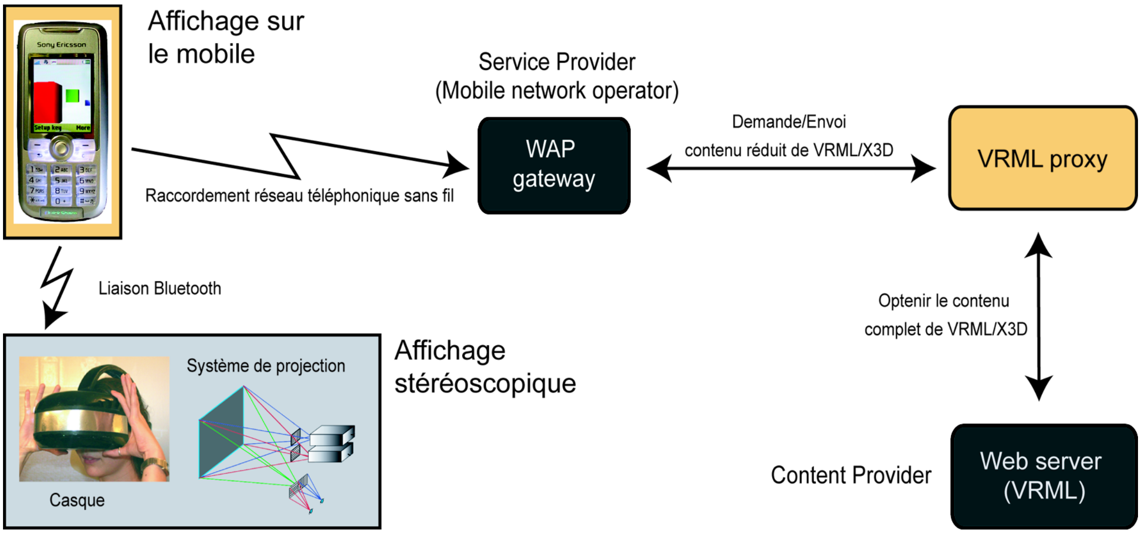


Figure 1: Vue d'ensemble de l'architecture du système

Cette transformation est exécutée par un "WAP gateway". Bien que GPRS utilise une transmission commutée par paquets, la connexion d'un appareil mobile se fait toujours via le "WAP gateway". La connexion TCP/IP s'établit entre le "WAP gateway" et le "proxy VRML" pour transférer les données, elles doivent être proches au sens des technologies réseau, afin de réduire au minimum les temps de latence.

Les possibilités d'optimisation des données de Réalité Virtuelle

Deux méthodes principales sont possibles pour l'optimisation des données de réalité virtuelle 3D. Les méthodes "d'optimisation statique" sont le pré-chargement, la mise en cache, la compression et la suppression des données inutiles. Le fait que tout ceci soit réalisé à la volée ne change pas les méthodes utilisées pendant ce processus, contrairement aux méthodes "d'optimisation dynamique" qui envoient des données partielles pour répartir équitablement la charge de travail entre le client et le proxy VRML. Le processus de ces optimisations peut être perturbé par le fonctionnement du réseau, mais une implémentation adaptée peut résoudre ce problème.

Ces méthodes ne sont pas indépendantes les unes des autres. L'équilibrage des charges entre le client et le proxy VRML mène à plus de communications et demande donc plus de bande passante. Dans tous les cas un équilibre optimal de toutes les méthodes utilisées doit être trouvé. Afin d'optimiser les calculs de rendu sur l'appareil mobile les niveaux de détails doivent être adaptés en réduisant le nombre de facettes des objets.

Le réduction du contenu

Un monde virtuel 3D est constitué de plusieurs objets géométriques (au sens défini par VRML). Ces objets décrivent la structure géométrique de la scène 3D. Le logiciel de visualisation permet à l'utilisateur de se déplacer librement dans la scène via les touches de navigation ou tout autre périphérique d'entrée. De cette façon l'utilisateur peut choisir n'importe quel point de vue pour observer la scène.

Dans tous les langages de description, des valeurs par défaut sont définies. Si les valeurs définies ne diffèrent pas de ces valeurs par défaut, elles peuvent être enlevées sans perte d'information. Dans la scène virtuelle 3D un contenu additionnel peut être inclus, comme du son ou des images. Si l'appareil mobile ne possède pas les capacités pour traiter ces informations (par exemple bruits de fond) il n'est pas utile de transmettre ces données. Cette information peut être enlevée au niveau du proxy VRML sans perte côté client.

Quand un appareil mobile demande un fichier représentant une scène 3D, l'appareil doit informer le proxy VRML sur les capacités de l'appareil: capacité de traitement, résolution de l'écran, capacité de gestion du son, etc. Côté proxy VRML le fichier spécifique est demandé à un serveur cible. En analysant ce fichier le proxy VRML crée une représentation de cette scène orientée objet. Des ressources additionnelles (par exemple des liens vers d'autres fichiers) peuvent être détectées et pré-chargées. Le proxy VRML enlèvera les informations inutiles et non gérées, selon les spécifications du mobile. La scène 3D est transformée de nouveau dans le format de données correct (dans ce cas-ci VRML) tandis que la méthode de réduction du contenu est prise en considération.

Une fonction complémentaire du proxy VRML est la conversion de données pour les appareils mobiles. Si, par exemple, un format n'est pas géré, il peut être converti par le proxy VRML. Cette fonction peut parfois nécessiter une bande passante plus importante, elle doit donc rester optionnelle.

La transmission incrémentale des données (Streaming)

Par l'utilisation de la transmission incrémentale des données toute l'information n'est pas envoyée: au départ, seuls les objets qui sont près de la position de l'utilisateur dans la scène sont envoyés au mobile et sont affichés. Si l'utilisateur se déplace à l'intérieur de la scène virtuelle 3D le client doit envoyer au proxy VRML la position et la direction de son déplacement. Selon cette information le proxy VRML peut recalculer la position de l'utilisateur dans la scène et envoie les objets additionnels au client si nécessaire. Dans ce cas la quantité de données transmise est diminuée parce que seules les données nécessaires sont transférées.

Pour éviter que des objets apparaissent soudainement en raison d'un temps de latence élevé et/ou d'une faible bande passante entre le client et de le proxy VRML le "déclenchement" de l'envoi de nouveaux objets doit être prédit.

Répartition des temps de traitement des données 3D

Coté mobile, la tâche critique est le calcul du rendu de la scène 3D. A partir des données 3D, le client doit calculer la position des objets, prendre en compte les lumières, les ombres, etc. C'est une tâche lourde pour un système qui ne présente que de faibles capacités de mémoire. Plus une scène est complexe, plus la capacité de calcul nécessaire pour traiter toutes les données en quasi temps réel est importante. Une solution possible à ce problème est de décharger une partie du traitement vers le proxy VRML.

Le client envoie une demande au proxy VRML de traiter une partie spécifique de la scène 3D. Après traitement le client récupère le résultat de la demande.

La visualisation stéréoscopique sur un appareil mobile

Une solution possible est la génération d'images anaglyphiques. Une image destinée à chacun des deux yeux est calculée en décalant les caméras correspondantes dans la scène. Une fois les images calculées, un filtrage des couleurs est opéré.

La composante cyan est enlevée de la vue gauche, la composante rouge de la droite ; ces deux images sont ensuite fusionnées. Observée avec des lunettes rouge-cyan, la scène est perçue en relief.

Une autre solution technique est d'ajouter sur l'écran de l'appareil mobile un filtre lenticulaire. Cette méthode ne fonctionnera malheureusement pas avec tous les écrans. Un avantage de cette méthode est que les utilisateurs n'ont pas besoin de porter de lunettes pour percevoir le relief. Ce dispositif dit auto-stéréoscopique nécessite qu'une colonne sur deux de l'image affichée corresponde à une vue destinée à chacun des deux yeux, ceci entraîne une baisse de la résolution de l'affichage de moitié.

La visualisation stéréoscopique avec un système à deux projecteurs vidéo

Le système de projection consiste en deux projecteurs vidéo où chaque faisceau projeté est polarisé. Dans ce scénario les données de réalité virtuelle sont envoyées par l'intermédiaire du mobile via une liaison Bluetooth à un ordinateur puissant qui fait le calcul des deux images destinées à l'œil doit et à l'œil gauche.

L'observation de l'écran de projection spécial à travers des lunettes polarisantes permet à chaque œil de ne voir que l'image qui lui est destinée.

Conclusion

Cette recherche permet à des utilisateurs de téléphones mobiles de visualiser des données de réalité virtuelle 3D. Du fait que ce projet est développé en langage Java, le logiciel fonctionnera sur tous les appareils mobiles compatibles.

Un aspect important est que les données 3D sont optimisées de manière spécifique pour chaque type d'appareil mobile, cette optimisation est assurée grâce à l'échange d'informations entre le client et le proxy VRML. Par conséquent des données 3D existantes peuvent être réutilisées sans nécessiter de réduire le contenu ou d'optimiser ces fichiers. Les fournisseurs de contenu n'ont pas s'occuper des différents types d'appareils mobiles ou d'offrir pour chaque appareil une version optimisée de leurs fichiers. Pour utiliser au maximum les capacités de l'appareil mobile, celui-ci doit pouvoir communiquer avec le proxy VRML. Si ce n'est pas le cas un sous-ensemble de ces fonctionnalités est pris en charge par le proxy VRML qui va par exemple optimiser on réduire le contenu de la scène 3D de manière transparente.

Avec l'aide de la transmission incrémentale des données par le proxy VRML une bande passante plus faible sera nécessaire.

Abstract (English)

3D Virtual Reality can often provide a realistic presentation of information to the user. With the help of description languages for Virtual Reality such as VRML, X3D, powerful means are available. Nowadays, many websites use this technology to provide the visitor with additional 3D information. As the high bandwidth of the Internet allows faster download, the complexity of such files can be increased by additional features. As a result, the client needs more processing power to parse these files. On personal computers this task is done by providing plug-ins for browsers. The task of this research project is to provide the users of mobile devices with the same 3D functionalities as available on personal computers. Mobile devices are limited in processing power, input and output capabilities. Although the memory in new devices is growing, most of them still have a low amount of main memory and either no or very little memory which can be used for persistence storage. Moreover, the connection via the wireless network is limited in bandwidth, has higher latency times and is not as reliable as a cable network. Therefore, the 3D Virtual Reality data has to be optimised before transmission over wireless networks. For this purpose the concept of the transparent "VRML proxy" was developed.

Several modelling languages exist to describe 3D Virtual Reality data. Out of these different formats the "Virtual Reality Modelling Language" (VRML) was selected. This is a well known and widely used format for Internet applications. Although there exists a newer version of the standard, which is called X3D, there is no difference between VRML and X3D with respect to the considered functionality. Since the work was started on the base of VRML, this standard will be the one mentioned and used.

Interest in the ability to view and manipulate 3D data on mobile devices has been shown by several research groups, with a variety of intended applications including entertainment, education or instant help.

Aim and Concept

The aim of this project is to add support for 3D Virtual Reality data on mobile devices and additionally to present this information in monoscopic as well as in stereoscopic view.

Similar to the process used to realise 3D Virtual Reality data on personal computers, there are three different possible locations for interaction with the original file content in order to optimise the content for presentation on mobile devices:

- At the server side
- At a proxy in the middle
- At the client side

In-depth research has already been done on the base of a "client proxy concept". What is important for the client side is further research on the capabilities of mobile devices and, for the proxy server (referred to as the VRML proxy), research on mobile telecommunication networks and 3D description languages (X3D, VRML).

System Overview

Figure 2 shows the overall system architecture that is used to connect the mobile devices. For mobile devices, several possibilities exist to connect to the Internet concerning the bearer technology. In Europe, the transmission can be established using GSM, HSCSD or GPRS and UMTS. EDGE systems are more commonly used in the USA as an intermediate step towards UMTS. Since GSM and HSCSD are circuit switched transmission, a transformation into the packet switched transmission of the Internet has to be performed.

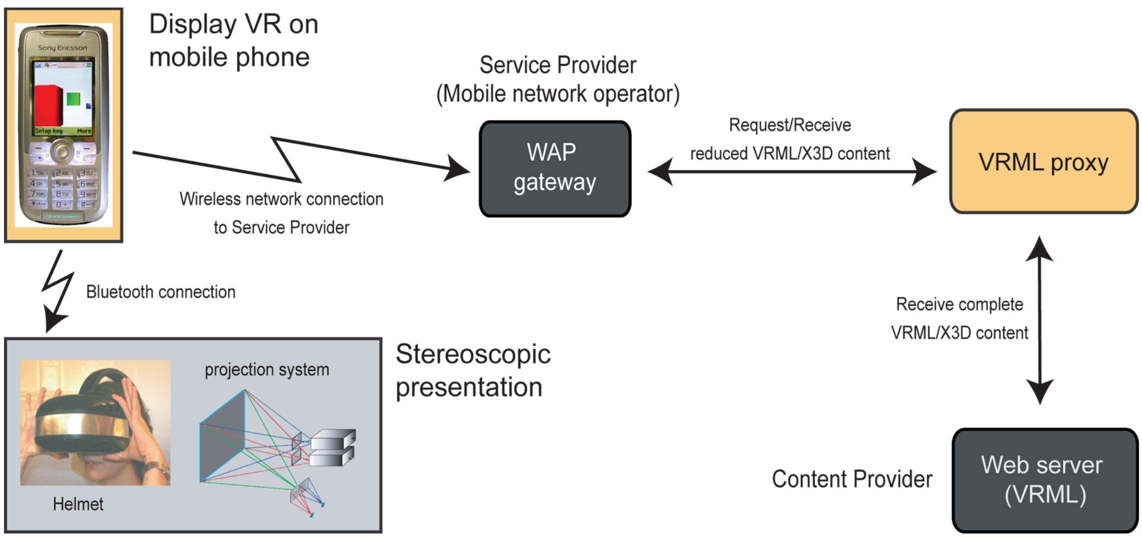


Figure 2 System overview

This transformation is performed by a “WAP gateway”. Although GPRS is a packet switch transmission, the mobile device connection is done using a “WAP gateway”. The reason is that the WAP protocol is optimised and encoded for the wireless network bearers. Therefore, every producer of mobile devices uses this well known technology to optimise the communication. Since a TCP/IP connection is established between the WAP gateway and the VRML proxy to transfer data, they should be adjacent in sense of network technology, in order to minimise the latency time.

Optimisation Possibilities for Virtual Reality Data

There are two principle methods for the optimisation of the 3D Virtual Reality data. The “static optimisation” methods involve preloading, caching, compression and removal of data. All this is “done on the fly”, but the methods themselves are not changed during the process. In contrast to the “dynamic optimisation” methods which send partial data or try to achieve a load balance between client and VRML proxy. However, during the execution of these methods, the network related conditions can alter. By an adaptable implementation of these methods, this problem can be solved.
These two methods are not independent of each other. Balancing more load between client and VRML proxy leads to more communication which needs more transmission bandwidth. In all of these cases, an optimal balance of all used methods has to be found.

In order to optimise the 3D image on the mobile device, the rendering strictness means that the amount of meshes which are used to present an object can be increased or decreased.

Content Reduction

A 3D Virtual Reality world, which is referred to as a scene, is usually built by more than one geometrical object (called node in VRML). These objects (nodes) describe the geometrical structure of the 3D scene. The software allows the user to freely move through the scene by navigating with the key pad, mouse or other input devices. In this way, the user can choose the viewpoint to look at the scene from any preferred angle.

In all description languages, default values are declared. If a value is identical to the default one, this value contains redundant information which can be removed without any loss of information.

In a 3D Virtual Reality scene additional content such as sound and images can be included. If the mobile device does not have the required capabilities to provide this information (e.g. play back sounds) it does not make sense to transmit the data over the wireless network. This information can be removed at the VRML proxy without any loss at the client side.

When a mobile device requests a specific 3D Virtual Reality file, the device has to inform the VRML proxy about the device capabilities, such as processing power, screen resolution, supported sound formats and so on. On the VRML proxy side, the specific file is requested from a target server. By parsing this file, the VRML proxy creates an object oriented representation of that scene. Additional resources (e.g. links to other files) can be detected and preloaded. The VRML proxy will remove unneeded and unsupported content according to the mobile device specification. The 3D Virtual Reality scene is transformed back into the correct modelling language (in this case VRML) while these methods are taken into account to reduce the content.

An additional feature of the VRML proxy is the data conversion for mobile devices. If, for example, a sound format is not supported, it can be converted by the VRML proxy. This feature can be used if enough bandwidth is available and the user wants to get this information. This feature has to be optional because the user has to pay the transmission.

Incremental Data Transmission (Streaming)

Through the use of incremental data transmission, a partial amount of information is sent rather than the maximum required amount of information. Initially, the objects which are close to the user's position in the scene are sent to the mobile device and displayed. If the user is moving inside the 3D Virtual Reality scene, the client has to inform the VRML proxy about the actual position as well as the direction of movement. According to this information, the VRML proxy can recalculate the user's position in the scene and send additional objects to the client if needed. In this case transmission bandwidth is saved because only necessary data are transferred.

To avoid the problem that objects are suddenly appearing next to the user because of high latency times and/or low bandwidth between client and VRML proxy, the “trigger” for sending new objects has to be observed carefully. For the reason that the delay time and especially the bandwidth are variable in modern wireless communication systems during the connection, this property has to be taken into account for the decision of when to send new data to the client.

Distribution of 3D Data Processing

The presentation of 3D Virtual Reality scenes is one of the most problematic tasks for mobile clients. According to the 3D Virtual Reality data, the client has to calculate the position of the objects, the lightning and shadows. This is a difficult task to perform with weak processors and low main memory. The more complex a scene is, the more processing power is needed to process all data in nearly real time. One solution for this problem is to perform tasks which require large amounts of processing power at the VRML proxy rather then at the client side.

The client sends a request to the VRML proxy to process a specific part of the 3D Virtual Reality scene. After the processing is completed at the VRML proxy side, the client receives the result of the request. Initially, this method may appear to be counter productive because, using this method, the amount of data which has to be transferred grows. This is true with respect to the overall data transmission, but the main goal is to optimise the view of 3D Virtual Reality data on mobile devices. Under the circumstance that the mobile device is too weak to “render“ the 3D Virtual Reality data in real time, it is advantageous to let the VRML proxy render the data and send back the result.

Stereoscopic View on Mobile Device

One technical solution to provide stereoscopic view on mobile devices is the generation of anaglyph images. Every displayed image out of a sequence of images is displayed by merging the image for the left eye and the image for the right eye. The green and blue colour components of the image are removed for the left eye and the red colour component of the image is removed for the right eye before the two images are merged. A disadvantage of this technique is that the users have to wear glasses with cyan/red filters.

Another technical solution is to add a lenticular lens array to the mobile phone screen. Unfortunately this method will not work with every mobile phone screen. An advantage of this method is that users do not need any additional equipment such as special glasses.

Stereoscopic View with a Two Video Projectors System

The projection system consists of two video projectors where each video projector beam is filtered with one polarised filter different in polarisation direction. In this scenario the VR data are sent via mobile phone over a Bluetooth connection to a powerful personal computer which does the calculation of the left and the right eye image scenes. The information for the left eye is sent to the first projector and the one for the right eye to the second projector. The stereoscopy is rebuilt by the stereo vision because the left and right eye information is filtered for the corresponding eye by the polarised filter. The movement in the Virtual Reality scene is controlled by the mobile phone over the Bluetooth connection.

Conclusion

This research enables users with mobile devices to view 3D Virtual Reality data. Since this project is based on Java, the software will work on every Java enabled mobile device.

An important feature is that the 3D Virtual Reality is optimised for the specific mobile device. This ability is assured by exchanging information between client and VRML proxy. Therefore, existing 3D Virtual Reality files can be reused without reducing content and optimising these files. Content providers do not have to care about the different mobile devices and to provide an optimised version of their file for every device. This fact is of particular importance if the 3D files are generated automatically.

To be able to use the highest possible amount of functionality, the mobile device has to be able to communicate with the VRML proxy. If this is not the case, a subset of the functionality can be used by redirecting requests for VRML files to the VRML proxy. In this case, the functionality is reduced to a “transparent” proxy which reduces content.

With the help of the incremental data transmission of the VRML proxy, low bandwidth systems can also be used. Data transmission is reduced and adapted to the actual low bandwidth system. This avoids the need of download the entire file to the device, which is a time consuming and costly method of transmission in an environment with low bandwidth.

1 Introduction

For hundreds of years, human beings have created illustrations of their environment in order to model and construct their surroundings. In modern computer systems, such three-dimensional illustrations are called Virtual Reality (VR). The least common denominator of a definition of Virtual Reality is: to *establish a computer-based simulation of a real or an imaginary world*. The user is able to interact in real time with this synthetically generated, three dimensional environment.

During the last ten years, Virtual Reality has gained more and more significance, even though the first development of Virtual Reality started in the middle of the 1950s. The reason for this gain in interest is that, on the one hand, the technology has gained more publicity through books and films (Tron, Neuromancer, Matrix, The 13th Floor) and, on the other hand, the technology has improved dramatically since its first development. Because of these technical improvements, Virtual Reality is more realistic and easier to use and thus more content became available. A modern VR system consists of several components [BEI04]:

- Input devices: keyboard, mouse, trackball, data glove, etc.
- Software
- Computer system(s)
- Display system: Standard CRT monitor(s), Head Mounted Device (HMD), projection screens, CAVE, etc.
- Sound output

The software has to coordinate the user interaction with the virtual environment. These interactions are performed with the help of the input devices. Some systems also perform additional tracking of the user. User tracking could be realised from position recognition up to head and eye tracking. The generation of the synthetic generated environment including the user's interaction and manipulation requires powerful computer systems to be able to perform the calculations of the Virtual Reality in real time. Thus, powerful graphic workstations are used which have the needed processing power as well as powerful graphic cards to perform the three dimensional rendering. The display of the Virtual Reality can be realised on a standard CRT or digital monitors. In order to provide a realistic stereoscopic view, standard CRT monitors, projection screens, CAVES in combination with glasses or HMD can be used.

The reason for the use and the benefits of Virtual Reality are [MIC96]:

- Manipulation in real-life would cause damage (e.g. crash test)
- Manipulation in real-life would be cost intensive (e.g. civil engineering, plant engineering and construction)
- Rapid prototyping (e.g. automotive industry)
- “Edu-tainment” - Education and entertainment (e.g. training, games)

The first branch of industry which discovered the profitability of VR-technology was civil engineering. Nowadays, this technology is rapidly growing in many branches of industry. Many industries realised that, with the help of Virtual Reality, it is possible to save time and money, as well as gain more usability and provide more services.

The VR technology was brought to the end user with the growth of the Internet. Since Virtual Reality can provide a realistic presentation of information to the user, it was included in many websites to provide the visitor with additional 3D information. With the help of description languages for Virtual

Reality (e.g. VRML, X3D), powerful means are available. As the high bandwidth of the Internet allows for fast download, the complexity of such files can be increased by additional features. A side effect of the increasing complexity of these files is the increasing processing power needed by the clients in order to parse and display these files. On personal computers, this task is performed by providing plug-ins for browsers.
All of the above mentioned systems are high-capacity systems, thus they are more or less space consuming and **not mobile or not wearable**. This is typically not a problem for industry purposes, in contrast to applications for standard users. For these users, mobility becomes more and more important and the need for these applications to be integrated into mobile or wearable equipment increase.

The motivation of this research project is to provide users of mobile devices with the same Virtual Reality functionalities of personal computers.

Mobile devices (PDA's, smartphones and mobile phones) are limited in processing power, input and output capabilities. These devices also have a low amount of main memory and either no or little amount of memory which can be used for persistence storage. Moreover, the connection via wireless network is limited in bandwidth, has higher latency times and is not as reliable as a cable network. Therefore, there exists a need to optimise the Virtual Reality data for mobile devices before transmission via wireless networks. For this purpose, the transparent “VRML proxy” was developed.
The concept of the “VRML proxy” discussed, described and developed in this document addresses solutions for the following problems:

- Preloading
- Caching content
- Reducing content for transmission
- Optimisation for mobile devices
- 3D data display

Figure 1.1 shows a system overview with all components needed for operation. The black components are existing components while the yellow and grey components are newly developed components. The grey components are developed in the context of the MediaPhtothics project (Chapter 6.4), a subproject of the RhenaPhotonics project. The yellow components are developed in the frame of this research project.

To evaluate the effectiveness of these components and the introduced optimisation methods, the constraints which they are referring to and the importance of these constraints in the overall system (Figure 1.1) have to be discussed and rated.

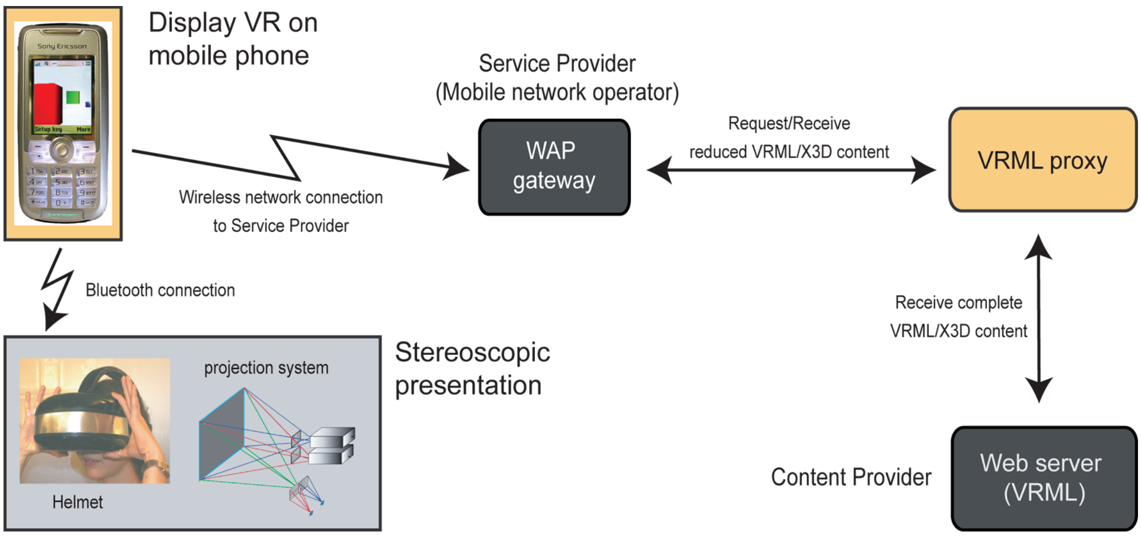


Figure 1.1: System overview of existing infrastructure

2 Telecommunication Systems for Mobile Devices

2.1 History of Mobile Telecommunication Systems (1G, 2G, 2G+)

The first analogue cellular telephone systems where developed after the Second World War. Users were able to phone, but only within their network cell. The hand over between cells was not yet implemented, therefore phone calls were dropped during attempted switches between cells.
Mobile systems finally became successful in the late 1970s and early 1980s but all developed systems were proprietary. With this development, they were able to locate the mobile phone and handle incoming and outgoing calls as well as hand over calls between different cells.
The most used "First Generation" (1G) systems are shown in Table 2.1 [RWO95].

System	***Used in***
Advanced Mobile Phone System (AMPS)	United States, Canada, Mexico, Australia, New Zealand, Taiwan, South Korea, Singapore, Hong Kong, Thailand, Brazil, Argentina
Total Access Communication System (TACS)	United Kingdom, Ireland, Spain, Italy, Austria, United Arab Emirates, Bahrain, Kuwait
Nordic Mobile Telephone (NMT) System	Sweden, Denmark, Norway, Finland, Belgium, Austria, France, Hungary, Netherlands, Spain, Turkey, Switzerland
A-Net, B-Net and C-Net	Germany

Table 2.1: Country list of former used 1G mobile phone systems

In the early 1980s, analogue cellular telephone systems were growing rapidly in Europe, particularly in Scandinavia and the United Kingdom, but also in France and Germany. Each country developed its own system, which was incompatible with the equipment and operation of any other system. This was an unsatisfactory situation because not only was the mobile equipment limited to operation within national boundaries, which in a unified Europe was becoming increasingly impractical, but there was also a very limited market for each type of equipment. Thus, economies of scale and the subsequent savings could not be realised in this situation.
Another driving influence to the development of the second generation systems was the development of new technologies in the semiconductor and microprocessor field. With these new technologies it was possible to realise digital systems.
The USA enhanced the AMPS into D-AMPS. With the prospective establishment of a European Union, the european countries decided to develop a common mobile communication standard. In 1982, the Conference of European Posts and Telegraphs (CEPT) formed a study group called the "Groupe Spécial Mobile" (GSM) to study and develop a pan-european public land mobile system.

The proposed system had to meet certain criteria:

- Subjective good speech quality
- Low terminal and service costs
- Support for international roaming
- Ability to support hand-held terminals
- Support for range of new services and facilities
- Spectral efficiency
- ISDN compatibility

The developers of GSM chose an so far unproven digital system in contrast to the standard analogue cellular systems such as AMPS in the United States and TACS in the United Kingdom. They believed that advancements in compression algorithms and digital signal processors would allow the fulfilment of the original criteria and the continual improvement of the system in terms of quality and cost. With over 8000 pages of GSM recommendations, they tried to allow flexibility and competitive innovation among suppliers and also provide enough standardisation to guarantee proper networking between the components of the system. This is realised by providing functional and interface descriptions for each of the functional entities defined in the system.

In 1989, the GSM responsibility was transferred to the European Telecommunication Standards Institute (ETSI) and, in 1990, the first GSM specification was published. Commercial service was started in the middle of 1991, and by 1993 there were 36 GSM networks in 22 countries. Although standardised in Europe, GSM is not only a European standard. Over 200 GSM networks (including DCS1800 and PCS1900) are operational in 110 countries around the world. In the beginning of 1994, there were 1.3 million subscribers world wide, which had grown to more than 55 million by October 1997. With North America making a delayed entry into the GSM field with a derivative of GSM called PCS1900, GSM systems exist on every continent, and the acronym GSM now stands for Global System for Mobile communications [RWO95].

In modern mobile communication systems, data services became more and more important. To meet these new requirements, de-/compression algorithms were continuously improved. Since these improvements mainly consisted in new software algorithms (2G+) or small hardware changes at the provider side, they were called the 2,5G.

The fact that the GSM family is the only important telecommunication standard for Europe concerning 2G and 2,5G networks, the following chapter is focused on the GSM standards. To give an overview of the next generation, the so called Third Generation (3G), the following chapter discusses the UMTS system(s) in general and the Wideband Code Division Multiple Access (W-CDMA) technology, which is particularly important for Europe, in more detail [HT01].

2.2 GSM Systems (2G & 2,5G)

2.2.1 Introduction to GSM Systems

The most important difference between the different GSM standards is the used carrier frequency band (for up- and downlink). Table 2.2 shows the used frequency band details for the three different GSM standards [RWO95].

Standard	***Details***
GSM 900 (Primary GSM)	Uplink frequencies between 890-915 MHz Downlink frequencies between 935-960 MHz It provides 124 channels with 200 kHz width
GSM 1800 (PCN 1800, DCS 1800) PCN = Personal Communication Network DCS = Digital Cellular System	Uplink frequencies between 1710-1785 MHz Downlink frequencies between 1805-1880 MHz It provides 374 channels with 200kHz width
GSM 1900 (PCS 1900) PCS = Personal Communication System	Uplink frequencies between 1850-1910 MHz Downlink frequencies between 1930-1990 MHz It provides 299 channels with 200 kHz width

Table 2.2: Frequency band details for GSM

The GSM 900 and the GSM 1800 are mainly used in Europe. Officially the GSM 1800 is known as the DCS 1800 standard. In the United States and Japan GSM 1900, officially called PCS 1900, is used.
In the evolution towards the third generation systems, the GSM systems were enhanced. These enhancements are:

- Intelligent network (IN) services
- Enhanced speech compression/decompression
- New transmission principles:
    - High speed circuit switched data (HSCSD)
    - General Packed Radio Service (GPRS)
    - Enhanced Data rates for GSM Evolution (EDGE)

While the HSCSD improvement of the GSM system needs mainly a software update, the GPRS improvement needs an additional small hardware change on the operator side. Extending a GSM system to an EDGE system, new hardware is needed. Most European operators have upgraded to GPRS and/or HSCSD, while in the USA the system operators are starting to build up EDGE systems.
A reason for this development in the USA is that GSM systems are actually almost not in use in the United States, thus the frequency ranges are unused in contrast to frequency ranges for UMTS which are occupied by systems still in use. Thus an intermediate step towards 3G is the introduction of EDGE. In Europe, the system operators are planning EDGE systems only in rural areas where an UMTS system would be to uneconomic [HH02, HT01].

2.2.2 Services of GSM Systems

The GSM network supports speech, data (fax) and a Short Message Service (SMS), which allows the user to send ASCII characters to another mobile phone

The network provides similar services as the ISDN network, which was a precondition for the network. The most important features of ISDN are:

- Identification (display coaling number, user profile, etc.)
- Knocking, halt the line, conferences
- Show the time/costs, data/time counter
- Data connections: GSM, GPRS/HSCSD, EDGE

2.2.3 GSM System Architecture

The three GSM standards use the same modulation (Gaussian Minimum Shift Keying, GMSK) and transmission algorithms (TDMA/FDMA). Thus they all have the same system architecture. All three standards use one frequency band for uplink (for mobile phone to the base station) and one frequency band for downlink (for base station to the mobile phone).

The area is divided into small cells. In neighbouring cells, different carrier frequencies are used so that it is almost impossible to have interference and/or disturbance or the interference and/or disturbance as a result of two or more carrier frequencies at the cell boundaries are to low to effect the transmission.

Since carrier frequencies for the GSM network are limited, the frequencies are “reused” in other cells. Between 7 and 9 cells are grouped together into a so-called cluster (Figure 2.1).

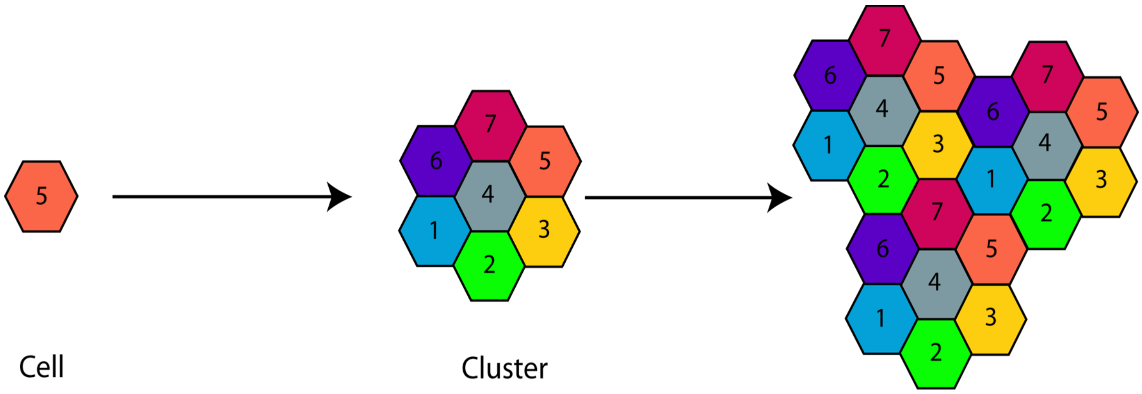


Figure 2.1: Grouping GSM cell to so-called clusters in order to reuse the frequencies

A mobile device, in general called Mobile Equipment (ME) needs the Subscriber Identity Module (SIM) to enable a connection to the network. The ME provides the hardware ability to connect to the network while the SIM contains the user information needed for identification (phone number), billing, etc.

The physical connection is established between the ME and the Base Transceiver Station (BTS, the antenna array). The BTS’s are connected to a so called Base Station Controller (BSC). The Base Station themselves are controlled by a Mobile Switching Centre (MSC). All information of a user is stored in the central Home Location Register (HLR) which is a database. The user information is authenticated via the Authentication Centre (AuC). If user are successfully authenticated the corresponding information is transferred to the Visitor Location Register which is a smaller database. In this database only the information of the users connected via this MSC to the core network are stored. The reason for the VLR is to optimise and speed up the database access which is needed to verify the connected user. Figure 2.2 illustrates the components needed for an operable GSM system.

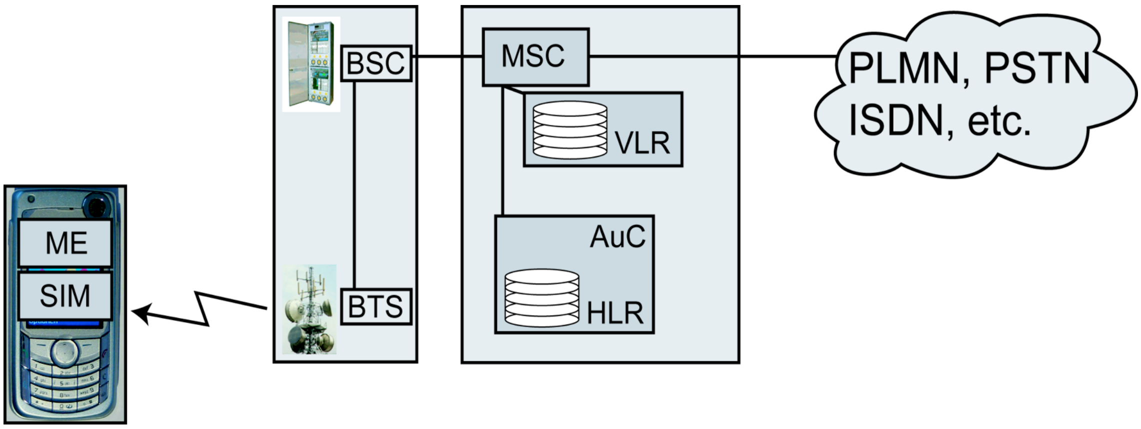


Figure 2.2: GSM system overview with the main important components

The MSC performs the switching and routing of the connections for incoming as well as for outgoing connections. The MSC can perform the connection between mobile devices connected to the same MSC as well as mobile devices connected to other MSC's (routing between MSC's). There are also interconnections between the MSC's and the local network which are realised with the help of gateways. This connection makes it possible to phone from the local network into any GSM network as well as to phone from a GSM network into another GSM network [RWO95].
The maximum speed for data connections depend on:

- Connection between ME and BTS
    - ME quality (antenna, power level, etc.)
    - BTS quality (antenna array, power level, amplifier, etc.)
    - Signal quality (interference, attenuation, etc.)
- Software of BSC
- Connection of core network
- Connection of gateway

2.3 The UMTS System

2.3.1 Introduction to the UMTS System

The Universal Mobile Telecommunication System (UMTS) is the upcoming Third Generation (3G) system. UMTS is a set of standards covered by IMT-2000 [HT01]. Common aspects are:

- Used world wide
- Used for all mobile communications
- Supports both packet switched (PS) and circuit switched (CS) data transmission
- Offers high spectrum efficiency
- Offers high data rates up to 2 Mbps (depending on velocity and actual load of the cell)
  e.g. W-CDMW:
  → 2 Mbps, maximal speed 10 km/h
  → 384 kbps, speed ~ 120 km/h

IMT–2000 is a set of requirements defined by the International Telecommunications Union (ITU). In total, proposals for 17 different IMT–2000 standards were submitted, 11 proposals for terrestrial systems and 6 for mobile satellite systems.

Evaluation of the proposals was completed at the end of 1998, and negotiations to establish a consensus among differing views were completed in mid 1999. All 17 proposals have been accepted by ITU as IMT–2000 standards.

The most important IMT–2000 proposals are the UMTS (W-CDMA) as the successor to GSM, CDMA2000 as the interim standard '95 (IS–95) successor, and time division synchronous CDMA (TD–SCDMA) (UWC-136/EDGE) as TDMA–based enhancements to D–AMPS/GSM, all of which are leading previous standards toward the ultimate goal of IMT–2000 [HT01].

2.3.2 Services of the UMTS System

The UMTS System provides the same services as the GSM systems. Additional features of the system are:

- Multimedia support
- Quality of Service Implementation (QoS)
- Location based services
- Virtual home environment

One of the most important new features mentioned above is the Quality of Service implementation. Table 2.3 shows the list of available QoS classes and features [UQOS, HT01].

Traffic class	***Fundamental characteristics***	***Delay***	***Example application***
Conversational class	Preserve time relation (variation) between information entities of the stream Conversational pattern (stringent and low delay)	Real time < 400 ms	Voice
Streaming class	Preserve time relation (variation) between information entities of the stream	Real time	Streaming video
Interactive class	Request response pattern Preserve payload content	Best effort	Web browsing
Background class	Destination is not expecting the data within a certain time Preserve payload content	Best effort	Telemetry, email

Table 2.3: List of Quality of Service (QoS) classes and the corresponding features

2.3.3 System Architecture of the UMTS System (W-CDMA)

In the first implementation step, the UMTS system is connected to the 2G core system. In the next step, all-IP-based core networks will be introduced. To enable handover and roaming, inter-networking gateways are required.

The UMTS network components are similar to the GSM network (Figure 2.3). In order to avoid confusion for the components of the UMTS networks another naming is used.

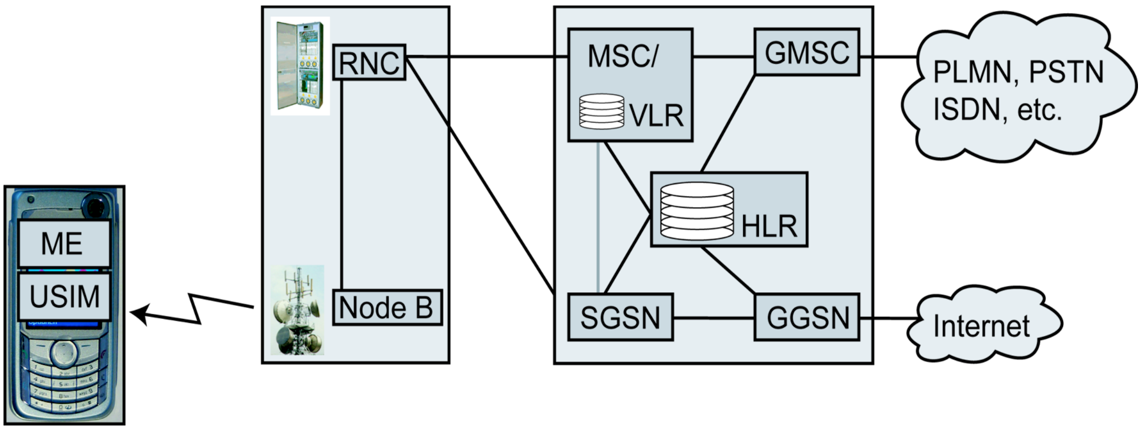


Figure 2.3: UMTS system overview with main important components

Table 2.4 shows the full expressions of the appreciations of Figure 2.3 and describes the similarity with the GSM systems [HT01].

Short cut	***Name***	***Description***
ME	Mobile Equipment	Hardware
USIM	Universal Subscriber Identity Module	SIM plus UMTS add-ons
Node B	Node B	BTS in UMTS network
RNC	Radio Network Controller	BSC in UMTS network
GMSC	Gateway MSC	New in UMTS
SGSN	Serving GPRS Support Node	New in UMTS, IP based
GGSN	Gateway GPRS Support Node	New in UMTS, IP based

Table 2.4: Short cut of UMTS components and the corresponding GSM commonality

2.4 WAP Standard

The abbreviation WAP stands for **W**ireless **A**pplication **P**rotocol. Since the most known application of the WAP specification is the micro browser, a common mistake is the confusion between WAP and the Wireless Markup Language (WML). WAP defines more than just the capability to “surf the Internet using mobile phones”. Another example of WAP standard capabilities is the so-called Push services, which are the basis for the well-known MMS service. Figure 2.4 shows a comparison between WAP and the Internet transmission (HTTP-based) [IWG03].

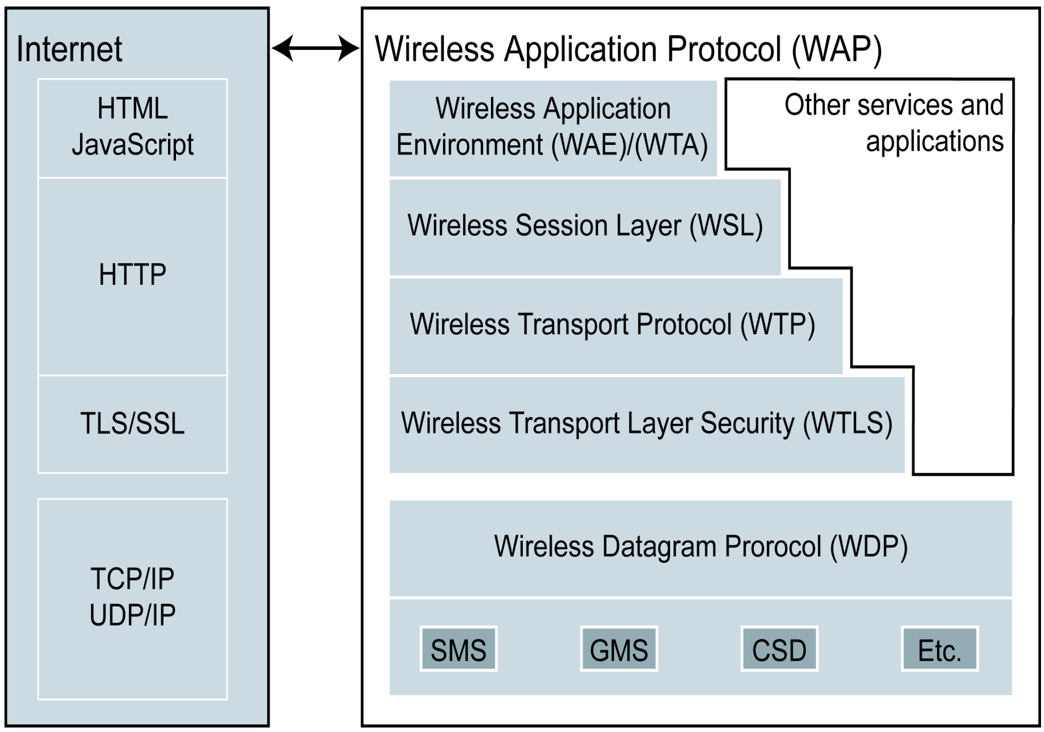


Figure 2.4: WAP protocol stack in comparison with Internet stack

The Wireless Application Environment (WAE) allows the execution of applications on the mobile device. The aforementioned most known application of WAE is the capability to “surf the web” using wireless devices. The Wireless Telephony Application allows applications to use and access the functionalities of the device (access phone book, accept/make phone calls) [HK01]. The WAP is independent of the bearer and can thus be used in different systems. To get a wide acceptance of the WAP, the specifications are reusing as many of the well-known internet protocols and standards as possible. Existing WAP standards are listed in Table 2.5 [WAP02].

Standard	***Important changes***
WAP 1.0 (1998)	Commercially unimportant (data over SMS)
WAP 1.1 (1999)	Use of CSD data transmission
WAP 1.2 (1999)	Push service, user agent profile
WAP 1.2.1 (2000)	Improvement
WAP 2.0 (2002)	XHTML, WSP, WDP and WTLS replaced by HTTP and SSL

Table 2.5: List of WAP standards and main important changes

Mobile devices which are requesting information via a WAP connection first have to connect to a so-called WAP gateway (proxy). A direct connection from the mobile device to the server via WAP (Version 1.x) is impossible, an intermediate gateway is always required. The reason for the intermediate gateway (proxy) is the fact that the request as well as the response is encoded and decoded into a binary format which is optimised for the wireless connection.

Figure 2.5 shows the general connection of a mobile device to the Internet [WAP02].

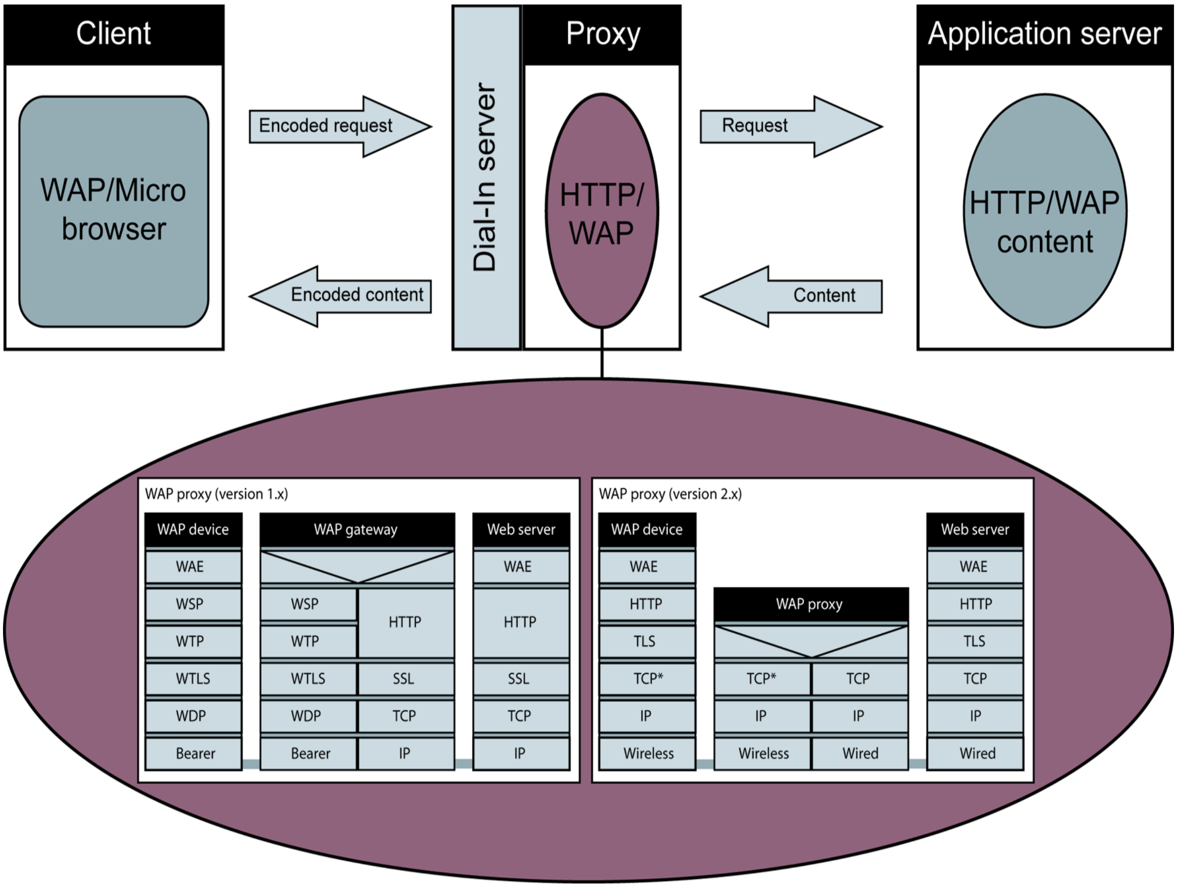


Figure 2.5: Internet connection of a mobile device, with a comparison between the two WAP standards, WAP 1.x (WAP gateway) and WAP 2.x (WAP proxy)

In the standard scenario shown above, the connection of a mobile device via WAP works in the following steps:

The mobile device connects to a Dial-In server where additional user authentication can be performed. The Dial-In server itself establishes the connection to the WAP gateway. The gateway does the encoding/decoding into a binary data format which is optimised for the over air transmission. Then the gateway forwards the request, sent from the mobile device, to the WAP (Web) Server. The requested WML (HTML) page is sent back to the gateway which performs the transformation into the optimised binary data. This data is sent back over the Dial-In server to the mobile phone [ED01, HEI00].

3 State of the Art of 3D on Mobile Devices

3.1 Introduction to 3D on Mobile Devices

As described in the introduction, the Virtual Reality is used to display 3D data. Such a Virtual Reality is usually called a virtual world or scene. For the purpose of describing such a world or scene, there are several description languages. The function of such a language is to define a specific amount of data which describe a 3D object in a scene. Well-known description languages are:

- VRML, VRML 2.0, VRML97, VRML NG or VRML99 (Virtual Reality Modelling Language), X3D
- M3G (Mobile 3D Graphics)
- 3DS (3D Studio)
- SDL, PovRAY (Scene Description Language)
- GDL (Geometric Description Language)
- XVL (eXtensible Virtual world description Language, XML based)

The importance of each description language depends on the application or area where it should be used. In addition, there exist several proprietary forms of 3D descriptions mainly used in games [WIL01].

In today's society, the trend to be more and more mobile is clearly visible. Thus the need for mobile devices to support 3D data presentation is increasing and will continue increasing in the future [BUR04, DON02]. To support this rising trend, one should first discuss which applications are useful on mobile devices. The following list shows what many research projects (see chapter 4.3) consider the most useful applications:

- Edutainment, games
- Product presentation
- Navigation
- Maintenance, surgery
- Search and rescue

For mobile phones and smartphones, 3D data viewers are not yet developed. The only exception is some PDA types, which meet certain criteria such as ARM, MIPS, SH-3 or Xscale Processor and Windows Mobile 2003, Pocket PC or Handheld PC 2000. The following Table 3.1 shows the devices which are supported by the Pocket Cortona client [PGPC].

	Xscale processor	***StrongARM processor***	***MIPS processor***	***SH-3 processor***
MS Mobile 2003	Supported	-	-	-
MS Pocket PC	Supported	Supported	Supported	Supported
MS Handheld PC 2000	-	Supported	-	-

Table 3.1: Hardware and software requirements for Pocket Cortona client

3.2 The Concept of Wearable Computers

In the last 30 years computers have developed from big and expensive machines towards small and relatively inexpensive personal computers. One key factor for this evolution was the introduction of semiconductors. Another push for the development of Personal Computer was the Internet and the increasing bandwidth to access it. In contrast to these two technical drives, the next evolution of computers is driven by a social factor: the mobility of users. Personal computers can be customised and configured by the user, hence they are in a certain way personal. Nevertheless, they are not integrated in the daily personal life tasks of the users, at least not without a lot of special effort and attentiveness from the user's side. Thus scientists and researchers developed the next generation concept, the concept of wearable computers.

In this concept, the user is the focus and not the wearable computer, thus the wearable computer has to be context and environment sensitive. It has to know where the user is and what he does, therefore the wearable computer has to be switched on. This fact additionally increases the usability because the device is immediately ready to use.

The wearable computer has to assist in daily life tasks. In order to fulfil this function, the device has to be carried by the user, thus it has to be small and light. In the ideal case, these devices can be so small as to be melted into clothing and thus really become "wearable computers" [MAN97].

An important difference of the wearable computer concept in comparison with other concepts is the user centred approach. An integral part of this approach is that users are expected to do something else while they are using the wearable computer, i.e. users are walking around and/or talking to others. To be able to handle these new criteria, the concept integrates the three operational modes for interaction into one, as shown in Figure 3.1 [MAN98].

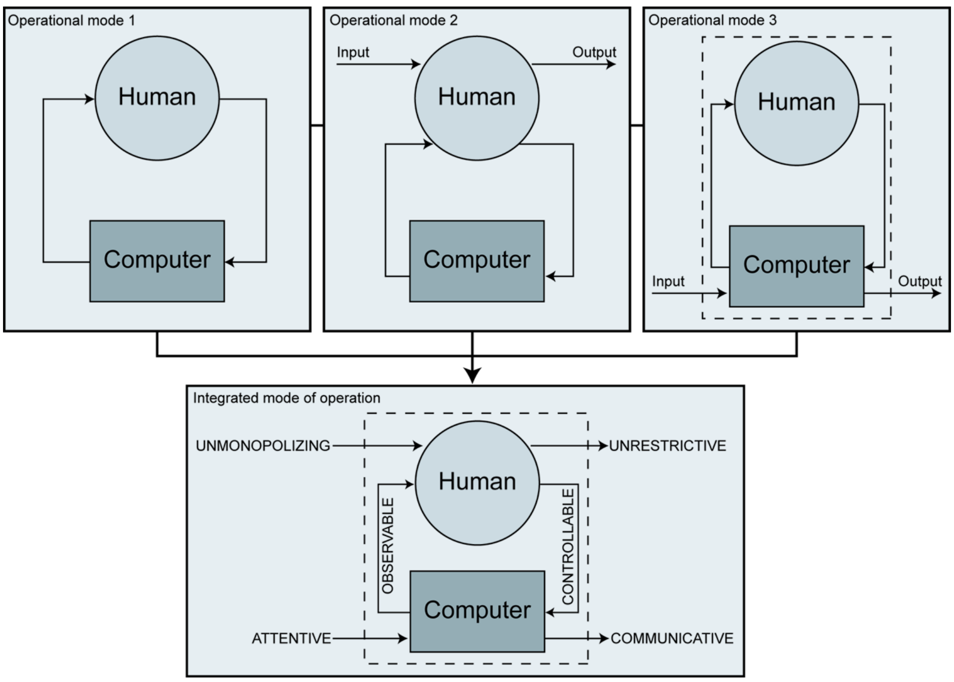


Figure 3.1: The tree operational modes of wearable computers, combined into one model

Out of this illustration the following six attributes of wearable computers are obtained [MAN98]:

- **Unmonopolizing** (interaction and control of the computer is a secondary activity, human being is the centre, melting with real world instead of cut off)
- **Unrestrictive** (user is mobile)
- **Observable** (user has to be able to always observe it, i.e. head mounted display)
- **Controllable** (user can get the control at any time)
- **Attentive** (the device is aware of environment, multi sensing)
- **Communicative** (communication medium)

As a result of this new operational mode of interaction, the “human–computer” interface has to be adapted to new input and output functionalities. The input has to be hands-free or at least almost hands free [ABO99]. Examples of hands-free input solutions are:

- Voice recognition
- Data gloves
- Eye/head tracking

In order to not attract too much of the user's attention, the output device has to be implemented as an augmented, half transparent display (Head Mounted Device), in the best case. In this case, the virtual world and information are melting with the real world. Therefore, the concept of wearable computer can be defined as:

- Small, light and part of clothing
- Always switched on
- Hands free user interface
- Merge output with real world
- Context and environment awareness
- Assist user, work in background, attract user attention only if needed

Applications of wearable computers, where these properties are needed to improve the performance and the usability of computing devices, include maintenance, search and rescue, surgery and education (edutainment) [MAN97]. These applications are highlighted in the following research projects.

3.3 VR Projects for Mobile Devices

The following is a list of research projects in the field of wearable computers which demonstrates some of the applications of this technology.

Project:	The Invisible Train
Category:	Game
URL:	http://studierstube.org/invisible_train
Description:	The Invisible Train is a mobile, collaborative multi-user Augmented Reality (AR) game, in which players control virtual trains on a real wooden miniature rail road track. In this project, wearable devices were used as thin-clients, while powerful (PC-based) servers performed the majority of the computations (such as graphics rendering). These virtual trains are only visible to players through their PDA's video see-through display, as the trains do not exist in the physical world. This type of user interface is commonly called the "magic lens metaphor". Players can interact with the game environment by operating track switches and adjusting the speed of their virtual trains. The game is synchronized between all participants via wireless networking.

Project:	SignPost
Category:	Navigation
URL:	http://studierstube.icg.tu-graz.ac.at/handheld_ar/signpost.php
Description:	The application guides a user through an unknown building by showing a variety of navigation hints, including a wireframe visualisation of the building structure superimposed on the video image (recorded by the device camera). An arrow points in the direction to go and a 2D overview map is available, too.

Project:	IPCity
Category:	Navigation, Maintenance, etc. (multiple purpose)
URL:	http://www.fit.fraunhofer.de/projekte/ipcity/index_en.xml
Description:	The research aim of the IPCity project is to investigate analytical and technological approaches to presence in real life settings. Analytically, this includes extending the approaches to presence accounting for the participative and social constitution of presence, the multiplicity and distribution of events in time and space. Technologically, this translates into developing portable environments for on-site configuration, mobile and light-weight mixed reality interfaces with the ambition to weave them into "the fabric of everyday life". Methodologically, this calls for moving "out of the lab" with field trials in real settings, applying a triangulation of disciplines and methods for evaluation. These range from interpretative-ethnographic to quasi-experimental approaches and include cognitive science, social-psychological and cultural-anthropological disciplines.

Project:	MARS (Mobile Augmented Reality Systems)
Category:	Navigation/Edutainment
URL:	http://www1.cs.columbia.edu/graphics/projects/mars/mars.html
Description:	The aim of the MARS project is to explore the synergy of two promising fields of user interface research: Augmented reality (AR), in which 3D displays are used to overlay a synthesised world on top of the real world and mobile computing. The increasingly small and inexpensive computing devices, which can be linked by wireless networks, allow the users to use computing facilities while they are roaming the real world. In exploring user interfaces, system software and application scenarios for MARS,

Project:	MARS (Mobile Augmented Reality Systems)
	the main focus is on the following lines of research: • Identifying generic tasks a mobile user would like to carry out using a context-aware computing system • Define a comprehensive set of reusable user interface components for mobile augmented reality applications. • Make combined use of different display technologies, ranging from head mounted devices, hand-held to palm-top in order to provide the best support for mobile users

Project:	PhoneGuide (Enabling Mobile Phones To Support Large-Scale Museum Guidance)
Category:	Navigation/Edutainment
URL:	http://www.uni-weimar.de/~bimber/research.php
Description:	Mobile phones have the potential of becoming a future platform for personal museum guidance. They enable full multimedia presentations and will significantly reduce acquisition and maintenance cost for museum operators if the users are using their own devices. However, several technological challenges have to be mastered before this concept can be successful. One of them is the question of how individual museum objects can be intuitively identified before presenting corresponding information. Enhanced museum guidance system uses widespread camera equipped mobile phones for on-device object recognition in combination with pervasive tracking. It provides additional location- and object-aware multimedia content to museum visitors and is scalable to cover a large number of museum objects. An object can be recognised and then related multimedia presentations such as videos, audio, text, computer graphics and images are displayed on the phone.

Project:	Kanji Teaching
Category:	Edutainment
URL:	http://studierstube.icg.tu-graz.ac.at/handheld_ar/kanji.php
Description:	This game-like application teaches players about kanji symbols that most ARToolKit users have already seen often but do not know what these kanji symbols mean. The novelty of this approach is that this project does not use regular workstations or laptops to host the AR (Augmented Reality) application. Instead, it use fully autonomous PDAs, running the application together with an optical marker-based tracking module that makes this application optimally mobile and available for a broad audience.

Project:	Virtuoso
Category:	Edutainment
URL:	http://studierstube.icg.tu-graz.ac.at/handheld_ar/virtuoso.php
Description:	Virtuoso is a collaborative educational game designed to showcase the possibilities of the hand-held Augmented Reality (AR) framework. The game's objective is to sort a collection of artworks according to their date of creation along a timeline drawn on a wall-mounted billboard. Every mark (fiducial) on the timeline carries one of the artworks, which are only visible through the player's AR PDA. The player can pick up any artwork with his PDA, by clicking on the artwork on the display and drop it on a free position by clicking on the free fiducial on the display. Since all positions are initially occupied, the game requires two or more players to cooperate in swapping or rearranging the sequence of artworks.

Project:	CONNECT
Category:	Edutainment
URL:	http://www.fit.fraunhofer.de/projekte/connect/index.xml
Description:	The European CONNECT project develops and uses leading edge information and communication technology to create an advanced learning environment, the virtual science thematic park. It connects 'informal' learning strategies and formal curricular activities in science education, involving students and teachers in extended episodes of playful learning.

Project:	iPerG
Category:	Edutainment/Game
URL:	http://www.fit.fraunhofer.de/projekte/iperg/index_en.xml http://www.pervasive-gaming.org
Description:	The iPerG project investigates the design of pervasive games and builds IT platforms for pervasive gaming. As a basis for creating attractive, economically viable pervasive games, iPerG develops methodology and tools to build and evaluate such games as well as concepts for their marketing and commercial exploitation. Using jointly developed platforms and methodology, the project will showcase different genres of games.

Almost all of these projects are focused on PDA's or even more powerful devices which can establish WLAN connections. This network connection is faster than a general GSM or UMTS connection and in most of the cases free or at least cheaper in cost.

The *PhoneGuide* project is an exception because it is based on the use of mobile phones and smartphones. The challenge in this project was to solve the problem of the object recognition and identification in order to transmit the multimedia content to the mobile device.

As in the *PhoneGuide* project, almost all of the projects have either pre-optimised content or they can rely on a fast network. For some of the projects, even both conditions are satisfied. None of the above mentioned projects have either the aim to optimise transmission via slower and more unreliable GSM/UMTS networks, or to optimise the 3D data display on low-end devices such as smartphones and mobile phones on the fly. Therefore this research work was introduced.

The aim of this research work is to optimise the 3D data transmission via GSM/UMTS networks and the 3D data presentation on the mobile device screen with the focus on smartphone and mobile phone applications.

3.4 Constraints and Features of Mobile Devices

3.4.1 Actual State and Future Perspectives of Constrains and Features

Presently, there exist three groups of mobile devices: the devices with higher capabilities such as PDA's, smartphones with medium capabilities and mobile phones which have in most cases low (or lower) capabilities. Table 3.2 shows a possible classification of the three devices types. The problem with this device classification is that all of these devices are being merged together. PDA's are starting to include phone functionalities and mobile phones are enhanced by PDA' functionalities.

	Mobile phone	***Smartphone***	***PDA***
Application area	Private	Private/Business	Business
Main application	Phone, SMS, MMS	Phone, SMS, MMS, Contact data	Phone, Contact data, Office (PowerPoint, Word, PDF)
Networking	GSM (UMTS)	GSM (UMTS, WLAN)	WLAN (GSM)

Table 3.2: Mobile device classification

During last years the mobile phone market experienced a rapid growth. Mobile phones have been improved with respect several aspects. One of these aspects is the improvement of the hardware such as the screen properties. An another aspect is the enhancement of the features of the mobile phones [BUR04]. A list of these enhancements to mobile devices is shown in Table 3.3. The most obvious hardware developments of mobile devices are the increasing processing power and the increasing size of main and storage memory [BEN, MPFB, NDF, SEDF].

	Mobile phone	**Smartphone**	**PDA**
Input	Small key pad, keys are used multiple times	Small key pad, some keys are reused	Larger key pad, key for every letter
Screen size	Small	Medium, some have touch screens	Relatively large, touch screens
Screen colour	Some, 256-4096 colours	Most, 256-65536 colours	All, 4096-65536 colours
Operating system	Manufacturer dependant, Symbian OS	Symbian OS, Windows Smartphone/ Pocket PC, Linux	Palm OS, Symbian OS, Windows Pocket PC, Linux
Heap size (Java)	512 KB – total memory	2000 KB – total memory	2000 KB – total memory
Total memory	3 MB – 80 MB sometimes extensible with MMC, SD and	12MB – 80MB most are extensible with MMC, SD and	512MB and more extensible mostly with CF Cards

	Mobile phone	**Smartphone**	**PDA**
	Memory Stick	Memory Stick	
Java	Most	Most	Third-party
Java3D	Some, rapidly increasing	Few	Almost non

Table 3.3: List of main important mobile device features

Even though the capabilities of mobile device are rapidly increasing, there are still important limitations to this development. One of the main important limitations is the battery capacity. As processing power increases, the energy consumption of the device also increases, which leads to shorter usage of the devices if the battery capacity is not increased. In most cases, more capacity results in larger battery size which is not desired for these devices as they are designed to be small and handy.

Future devices are trying to overcome these limitations. While the solutions to by-pass the problem of memory and processing power are relatively easy to obtain, the solution for input and output capabilities need new ideas. Examples of such new ideas are:

For input:

- Wireless keyboards (Keyboard connected with Bluetooth)
- Foldable keyboard
- Virtual keyboards (Device projects a virtual keyboard to the table)
- Data gloves

For output:

- Half transparent glasses
- Projection screens (Device projects the screen against a wall)
- Head mounted devices (HMD)
- (3D screens)

3.4.2 Theoretical Bandwidth and Delay Times

Originally GSM networks were introduced for voice communication, thus the data transmission was an added service for these networks. Initially, GSM networks started with a maximal data transmission of 9,6 Kbps. With the use of different codecs (error correction) the data transmission can be increased, in the best case, up to 14,4 Kbps.

Another improvement of GSM networks is the High Speed Circuit Switched Data (HSCSD), where more GSM time slots are used at the same time. From operator point of view, a disadvantage of this technology is the occupation of GSM channels even during idle time of data transmission because HSCSD reserves the time slot exclusively. Another disadvantage is the fact that this transmission is circuit switched and not packet oriented as the Internet (TCP/IP). Therefore, a transformation from circuit switched data into packet oriented data has to be performed. Due to the underlying GSM technology, the HSCSD speed for the up- and downlink is a multiple of 9,6 Kbps and 14,4 Kbps respectively. The theoretical use of up to eight time slots, with a resulting maximum data transmission of 115,2 Kbps, is actually not offered by service providers. Thus, the data rate for the transmission are often denoted by d+u where d is the multiple for the slot data rate for the downlink and u is the multiple of the slot data rate for the upload. For example: with 9,6 Kbps slot data rate, the denotation of 3+2 would mean a maximum data rate of 28,8 Kbps for downlink and 19,2 Kbps for uplink. Table 3.4 shows the available data transmission speed [RWO95].

Time slot(s) for Up-/downlink	***Downlink (Kbps)***	***Uplink (Kbps)***
2+1	28,8 (19,2)	14,4 (9,6)
2+2	28,8 (19,2)	28,8 (19,2)
3+1	43,2 (28,8)	14,4 (9,6)
4+2	57,6 (38,4)	28,8 (19,2)

Table 3.4: HSCSD timeslots combination and resulting transmission speed for up- and downlink

A third improvement of the GSM network is the introduction of the General Packet Radio Service (GPRS). This technology uses several GSM channels, too. In contrast to HSCSD, the GPRS technology is packet oriented and does not occupy the GSM channels all the time. GPRS occupies GSM channels only for several time slots during data transmission. During the idle time of data transmission, the GSM channel can be used by any other GPRS connection. According the GPRS standardisation, four different coding schemes per time slot are available (C1, C2, C3 and C4). Table 3.5 shows the data rate per time slot for the different coding schemes [RWO95].

Coding scheme	***Data rate (Kbps)***
C1	9,05
C2	13,4
C3	15,6
C4	21,4

Table 3.5: Resulting data rate per time for different coding schemes

GPRS additionally defines so called multislot classes to define the combination of up- and download settings.

The available classes are listed in Table 3.6 [RWO95].

Multislot Class	***Uplink slot(s)***	***Downlink slot(s)***	***Max. active slots***	***Time slot for uplink/downlink***
1	1	1	2	1+1
2	2	1	3	2+1
3	2	2	3	2+1, 1+2
4	3	1	4	3+1
5	2	2	4	2+2
6	3	2	4	3+1, 2+2
7	3	3	4	3+1, 2+2, 1+3
8	4	1	5	4+1
9	3	2	5	3+2
10	4	2	5	4+1, 3+2
11	4	3	5	4+1, 3+2, 2+3
12	4	4	5	4+1, 3+2, 2+3, 1+4

Table 3.6: GPRS multislot classes with resulting up-and download slots

Example: A GPRS class 6 device with C2 coding scheme can handle 3+1 or 2+2 data connections. The resulting maximum uplink data rate is 40,2 Kbps and 26,8 Kbps, respectively. For the downlink the data rate is 13,4 Kbps and 26,8 Kbps, respectively.

3.5 Measured Results and Benchmarks

For GSM the maximum data rate depends on the mobile device as well as on the service provider. Available data rates are 9,6 Kbps (typical) and 14,4 Kbps. HSCSD is not offered by every service provider. In Germany HSCSD is offered by D2 (Vodafone) with 38,4 Kbps and E-plus 56 Kbps. The GPRS service is offered by D1 (Deutsche Telekom), D2 (Vodafone), O2 and E-plus with 53,6 Kbps.
The above mentioned numbers were obtained from [EPG, O2G,TG,VG]. The difference of the maximum theoretical bandwidth and the offered one can not be answered by the author with *security* because the Service Providers are not providing any information. A reason could be limitations in the operators network as well as rounding because of marketing reasons.
The development of an own network benchmark was performed because of the unavailability and the missing documentation, respectively [JLT03]. The following mobile phone and smartphones where tested:

- **Sony Ericsson K700i, GPRS Class 10**
- Sony Ericsson P800/P900
- Nokia 6230
- **Nokia 6680, GPRS Class 10**
- Nokia N90
- MDA Compact
- Siemens SX70

Table 3.7 shows as example the measured results of two different mobile phones (Sony Ericsson K700i and Nokia 6680). Detailed description and the results are available at http://isign.fh-offenburg.de/vr-project/benchmark. The measured time is the duration between request and response of the server. The content length of the request is zero and the content length of the response is 2058 characters.

Phone	***Service Provider***	***Network Operator***	***GSM (9,6 Kbps)***	***GPRS (53,6 Kbps)***	***HSCSD***	***UMTS (unknown)***
Sony Ericsson K500i	D1	D1	2,3 sec	1,5 sec	-	-
Sony Ericsson K500i	Jamba	D1	2,5 sec	-	-	-
Sony Ericsson K500i	D2	D2	2,3 sec	1,5 sec	-	-
Nokia 6680	D1	D1	2,3 sec	1,34 sec	-	640 ms
Nokia 6680	Jamba	D1	2,6 sec	-	-	-
Nokia 6680	D2	D2	2,2 sec	1,13 sec	-	610 ms

Table 3.7: Benchmark results for different mobile phones

3.6 Implementation Constraints

3.6.1 Hardware Constraints

In order to handle a hardware constraint, one has to determine the mode of operation and the reason for the limitation. The most important hardware constraints are:

- Memory and persistent storage
- Processing power
- Big diversity of hardware
- Screen/keyboard

The limitation of memory and processing power is a problem which has to be solved by the manufacturers of such devices. However, in this work these problems will indirectly be addressed by taking them into account in the software development process.

The problem with the big variety of hardware is actually a problem of the underlying operation system. Nevertheless, the device diversity has to be kept in mind during the software development. In the worst case, it could mean that the software will be device specific.

The problem of the input limitation has to be taken into account during the software development, too. One has to take the least denominator of all devices which should be supported. New ideas such as wireless keyboards, data gloves, etc. are in this work not used because the application does not need such a big variety of user interactions.

The requirements were limited to eight different keys:

- Rotate left/right
- Move up/down
- Move left/right
- Zoom in/out

The requirements of at least eight different keys is fulfilled by every mobile device. The problem of the output screen is addressed by the use of built-in screens and the use of external devices.

All hardware constraints mentioned above have to and can be solved by software adaptation.

3.6.2 Software Constraints

Most software constraints are a result of the device diversity. As mentioned above, the device diversity results in a diversity in operation system architectures and subversions. Widely used system architectures and subversions are:

- Microsoft Windows:
    - MS Handheld
    - MS Mobile
    - MS PocketPC
    - MS Windows CE
- Symbian OS (actually v6.0 – 9.1)
- Palm OS (different versions)
- Psion
- Linux (embedded)

In this work, another software constraint is the availability of 3D software libraries. Trials to implement 3D software libraries have shown that this task is too time consuming, particularly if (free) libraries are available. At the beginning of this research work there were no free libraries available, therefore the development of such a library was started. However, during that time, mobile phone (smartphone) manufacturers realised the possible potential of 3D on mobile devices and started implementing libraries. Since 3D libraries are now available [NDF, SEDF], the development of a proprietary library was stopped.

There are several programming languages such as C/C^{++}, Java, Visual Basic, Fortran, etc. from which one has to be chosen. When deciding which one to use, advantages and disadvantages have to be taken into account. Because of the popularity and wide usage of C/C^{++} and Java, only these two languages are considered [GCC].

Table 3.8 compares the most important pros and cons for these two languages with respect to mobile device applications.

	C/C++		***Java***	
	Pro	***Contra***	***Pro***	***Contra***
Execution speed	Compiled => fast			Byte code => slower
Pre requirements		Higher	Low	
Portability		Medium/low proprietary API's	High standard API's	
Interpreter	Not needed			Virtual Machine
3D libraries		Not freely available	Freely available or built in	

Table 3.8: Comparison of C^{++} and Java according to mobile devices

Since Java has less pre-requirements, a higher portability and freely available 3D libraries, this language was chosen. Detailed research has shown that the optionally available built-in 3D library is the best choice. The requisite of this 3D library limit the portability but increase the operational speed because this library is hardware optimised.

4 Constraints and Solutions for Transmission and Display

4.1 Introduction to Optimisation Methods

In this context, optimisation means the process of improving a system in certain ways to increase the effective execution speed, the need of bandwidth and/or to reduce memory requirements. An optimisation method is to find optimal solution to a problem with respect to a specific aspect. Therefore, it can happen that an optimal solution with respect to one aspect of a problem is counterproductive with respect to another aspect of this problem. To solve this dilemma, one has to define an algorithm which makes a decision between the different aspects of the problem. These decisions can be human predefined or machine computed. In the case of an automated machine decision, this process results in a newly generated optimisation process.

Optimisation methods can be applied to improve:

- Application code to limit memory usage
- Application code to speed up execution
- Application code to minimise bandwidth and data transmission
- Application data to minimise transmission
- Application data to meet the needs of the application

To limit the amount of memory usage, the software has to be well structured and designed. The more unnecessary or temporary information is used, the more memory is consumed. In addition, the representation of the information directly corresponds with the amount of memory. For example to store the red value of a colour (range is from 0 to 255), it is not necessary to use a “double” variable to store the value. As a result of the growing main memory available on most devices, programmers do pay less and less attention to such facts. Following this common trend, this fact was not taken into account in this work as well.

To speed up execution, the programming language with the best performance on the device has to be selected. There are two programming language principles. With the first type of language, the human-readable source code has to be translated into device dependent machine code to be executed on this device. Typical examples are C/C^{++}, Fortran, etc. The second type of language makes use of an interpreter and therefore needs an additional “program”. Typical examples for such languages are Basic, Java, etc. In order to optimise the execution speed of the program, it is preferable to use a programming language which is of the first type.

Minimising the need of bandwidth and the overall data transmission are two important aspects because they are directly related with costs. In the case where the application needs a fast connection (higher bandwidth), the user needs equipment which can handle fast connection. Additional a fast network connection has to be offered by the network operator. Such a connection will be more expensive the faster it is. However, optimising the application code is not the only fact which influences the bandwidth and the data transmission.

Another fact is the type of application data. The minimisation of the delay time for data transmission in the application code reduces the need for bandwidth while compression can address both aspects.

Compressing the data reduces the overall data transmission as well as the need for higher bandwidth because more data can be transferred in the same time. The influence of the application with respect to the bandwidth and the data transmission is more difficult to optimise and under certain circumstances impossible.
Optimisation is impossible if an application need a specific amount of data in a specific time which is not offered by the network. Such an application is inoperable under such conditions.
Nevertheless, the data transmission can be optimised by minimising the amount of data transmitted to the client. This minimisation implies that all redundant data is removed. This can be achieved by defining default values for all attributes. Instead of transmitting these values, they can be omitted without any loss of information.
Improving data to meet the needs of the application will remove processing overhead. The data can be preprocessed so that processing and computing time at the client side is saved. This possibility plays an important role during the generation of the application data because during this process the client device ability can be taken into account in order to minimise processing overhead.
In order to be most efficient, one has to eliminate or at least decrease the effect of the most limiting constraints. To make the decision of which optimisation methods could be used, it is important to know the constraints of the whole system.
In the following chapter, different ideas of specific optimisation methods are discussed in more detail. This will help to understand the reasons for the decisions of which method was implemented and tested for a particular application. The optimisation methods can be divided into two different categories. The first category consists of methods which are applicable at the server side, i.e. before the transmission. Examples are:

- Data compression
- Data reduction
- Preload and cache
- Streaming (in connection with client)

The second category consist of methods which are applicable at the client side, i.e. after the transmission. Examples are:

- Data decompression
- Reduction of visualisation data
- Post load
- Streaming (in connection with server)

4.2 Transmission Time Reduction through Data Compression

In the previous chapter the relation between data amount, transmission time and bandwidth was explained briefly. However, this relation is important for the understanding of this chapter and therefore it will be discussed in more detail.
In order for interactive applications to function in an appropriate way, the response or reaction time of these applications have to be adequate. As discussed in the previous chapter, the answer of the question of what is an adequate value significantly depends on the person which is using the application. However, the general statement that the shorter the response or reaction time the more

convenient it will be for the user, is valid for all persons. Therefore, the shortest response or reaction time has to be achieved. Figure 4.1 illustrates the dependencies between data amount, transmission time and bandwidth related to data transmission of networks.

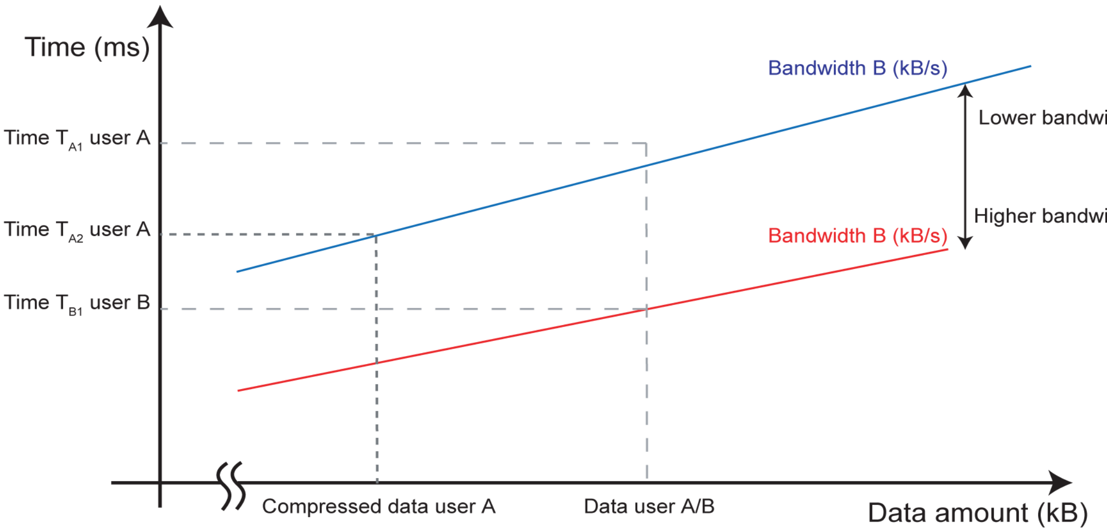


Figure 4.1: Relation between data amount, transmission time and bandwidth

In this example, there are two users with different bandwidth. User A has the lower bandwidth A and user B has a higher bandwidth. To download the same amount of data, user A needs more time (T_{A1}) than user B (T_{B1}). If the download time of user A should be decreased then either the bandwidth has to be increased or the amount of data has to be decreased. Because increasing the bandwidth is not always possible, one has to decrease the amount of data which can be achieved by compression methods. There exists two principle ways of compression: lossless and lossy compression. Lossless compression algorithms remove redundant data, therefore there is no loss of any kind of information. The algorithms discussed in the following are lossless methods.

The theory of data compression was developed in the paper “A Mathematical Theory of Communication” by Claude E. Shannon [SW63]. Shannon proved that there is a basic limit to lossless data compression, called entropy rate and denoted by H. Data can be compressed in a lossless manner with a compression rate close to H but it is mathematically impossible to achieve a better compression than H. Therefore, the best possible lossless compression rate is equal to the entropy rate. Lossless data compression theory is also known as source coding theory. This theory sets basic limits on the performance of all data compression algorithms. The theory itself does not specify how to design and implement compression algorithms. However, it provides some hints and guidelines on how optimal performance can be achieved [GZIP]. The most popular compression methods are the Lempel-Ziv methods (LZ , LZ77, LZ78), with the existing variation such as Lempel-Ziv-Welch (LZW), DEFLATE and Microsoft's CAB (LZX). The DEFLATE algorithm, which is a combination of the LZ77 algorithm and Huffman coding, is thought to be free of patents and is therefore widely used [RFC1952]. The name GZIP is an abbreviation for GNU ZIP, a GNU free software file compression utility designed to replace COMPRESS. In comparison with COMPRESS the GZIP method has much better compression rates and it does not use patented algorithms. It was adapted by the GNU project and is now relatively popular on the Internet. GZIP was written by Jean-loup Gailly and Mark Adler for the decompression code. The GZIP algorithm consisting of a 10 byte header, a body and an 8 byte footer. The body consists of the data compressed by the DEFLATE algorithm. Except for the header and footer, there is no difference between GZIP and DEFLATE. Table 4.1 shows a list of compression applications and the compression algorithm they are using [DCTC, GZIP].

Application	***Named as***	***Compression algorithm(s)***
WinZIP	PKZIP	LZ77, Huffman coding
PKZIP	PKZIP	LZ77, Huffman coding
Winace	ACE	Proprietary LZ
WinRAR	ZIP, LHA, CAB	LZ, LHA, LZX
Compress	COMPRESS	LZC
GZIP	GZ / GZIP	LZ77 and Huffman coding

Table 4.1: Compression applications and used compression algorithm

The achieved compression level does not depend exclusively on the used compression algorithm; it also depends on the data source. Table 4.2 shows average compression rates depending on the data source. Generally, the more similarities exist in the data source the higher is the compression rate [DCC].

Data source	***Compression rate***
Text data	60% - 98%
Binary data	30% - 75%
JPEG/PNG	-10% - 3%

Table 4.2: Average compression rate for different data sources

Image data are hard to compress because they are already pre compressed. In the worst case, some compression algorithms even lead to a higher data size than the uncompressed data. The decision of which algorithm to use for compression does not only depend on the achieved compression rate. The other aspects for this decision are:

- Compression speed
- Patent free
- Available libraries

In this work GZIP will be used because of the following reasons: Patent-free algorithm, free available libraries and good compression rate.

4.3 *Transmission Time Reduction through Data Reduction*

Removing data from the original data source can have two consequences:

1. No loss of information
2. Loss of information

In the first case, one removes information which can be regenerated. It represents information already known by the client because it is equal to a default value or known from data transferred previously. In the case these data are equal to default values, the information can be reduced without loss of information. This is shown in the following example. In this example, the original amount of data (red) can be decreased by more than 85% (green) without any loss of information.

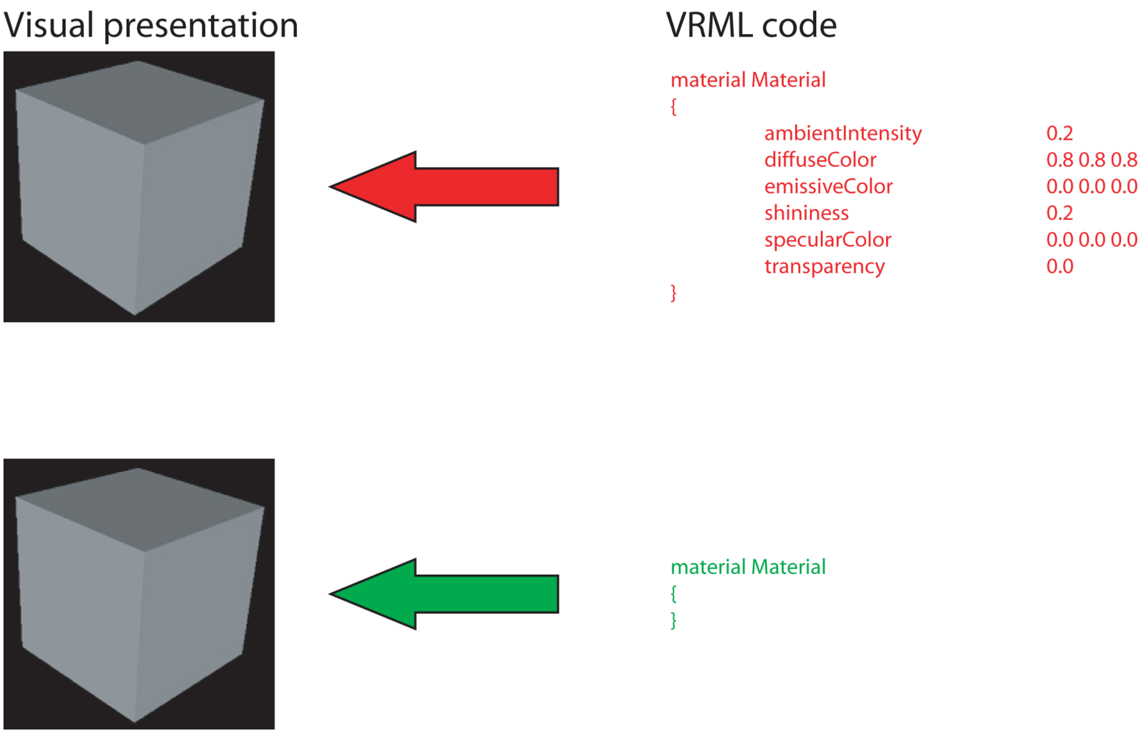


Figure 4.2: Red and green VRML code results in same visual representation, therefore VRML code can be reduced

The second case of reducing data can be treated as a lossy compression. In this case one removes data representing information. In order to avoid loss of content, only irrelevant data are allowed to be removed. The question which data is irrelevant and which one is relevant depends on the application where the reduction is applied. Furthermore, all data not available to the user due to device or client application weaknesses, can be assumed to be irrelevant. This is true even if they are carrying information important for the user. To explain this, consider a application where a VR scene contains an additional audio file which provides important information. However, since the device does not support audio output, this information is irrelevant and can be removed. Beside the device weakness, the client implementation with possibly limited features can play an important role.

In the context of the Virtual Reality data transmission, any object which visual presentation is too small

to be visible can be assumed to carry no information and is therefore irrelevant data.
In order to obtain the shortest transmission time, all unsupported contents have to be removed from the data source. Unsupported content because they carry at the actual state no information or rather the information can not be presented to the user. Moreover, it can happen that the user is not interested in a specific object or feature. In this case the information representing this object or feature can be omitted. Since the user does not want this kind of information, there is no loss if the information is not sent.

4.4 Delay Time Reduction through Cache and Preload

In order to decrease the amount of data to be downloaded, data are cached. To do so, these data are saved on the client side. When the information is requested, the client checks if the stored information are up-to-date. If this is the case, the client application will skip the download and present the locally stored information. Otherwise, an up-to-date version is downloaded. In most cases, the difference in version is identified by the date.
The preloading mechanism is counter productive to the caching mechanism in sense of decreasing the payload. The advantage of preloading data is that the delay time between requesting the information and presenting it is minimised because the information is already available for the application.

4.5 Optimising Transmission via Data Streaming

Data streaming, or simply streaming, is a technique to transfer a file in a continuous flow over the Internet. Therefore, the corresponding application can use the file before it is entirely received. Streaming technology is usually used for audio and video data files which are additionally compressed. This is in contrast to a download, where the entire file must be received before it can be used. Typically in case of streamed data, the information is not saved on the client device and cannot be reused later. In multimedia applications, streaming is considered as a “just in time” presentation of the content. A server delivers blockwise content to a client, presenting the data. A problem of streaming technology is the unreliable network (Internet) between client and server. Due to missing or ineffective implementations of Quality of Service (QoS), a client or a server cannot reserve a certain amount of bandwidth. As a consequence, the client can receive more data than he needs at a certain time and in the next moment he can receive too less. To overcome this problem, most clients are using a so-called buffer to save the data temporarily. In the case that the client receives too much data, the data can be saved in the buffer. In the case that the client receives not enough, he can still read data out of the buffer. In the case that the buffer is empty and the client receives not enough data, the presentation of the data will not be smooth. An advantage of streaming technology is that the user can begin listening or viewing the file, often within just a few seconds, whereas downloading could take much longer, depending on the speed of the connection and the size of the file. Additionally streaming technology is a solution to the problem of audio and video broadcasts where live audio or video is transmitted over a longer period of time. The difference between streaming and downloading is similar to the difference between watching TV and watching a video. Another example of streaming implementation is the so-called interlaced technology available in the GIF image format [GIFS]. The image data are ordered in such a way that the first information gives a rough impression of the image. All further data improve the overall quality of the image until all data are loaded. Table 4.3 lists available audio and streaming data formats or protocols [JRS, MEN02, RSTP, TOP02].

Name	***Format***	***Protocol***
Real Audio (RA)	Audio	HTTP, RTP/RTSP, NSV
MP3	Audio	HTTP, RTP/RTSP, NSV
OGG	Audio	HTTP, RTP/RTSP, NSV
Windows Audio (WMA)	Audio	HTTP, RTP/RTSP, MMS, NSV
Motion JPEG	Image/video	HTTP, RTP/RTSP, NSV
Windows Video (WMV)	Video	HTTP, RTP/RTSP, MMS, NSV
MPEG-4	Video	HTTP, RTP/RTSP, MMS, NSV

Table 4.3: Audio streaming formats and corresponding protocols

In standard situations the content of a file is completely transferred to the client. The client parses the content and performs a visual representation of the data. In case of “streaming VRML” the content is send partially. In order to achieve an advantage of the streamed information, the data has to be ordered. The objects and extended information have to be arranged in such a way that they are ordered according to their visual importance.
Objects are ordered according to the distance from the initial view point in relation to their size. Textures and other add-ons are queued to the end or removed. Of course, there is a small communication overhead. However, this overhead is negligible.

4.6 Extension of the Client–Server Concept with a Proxy

As a result of all the features discussed above, the standard client-server concept has to be modified. The modifications are caused by the need for extra intelligence for the optimisation processes. In order to optimise the data transmission for the mobile client over the wireless network, there are three principle locations where the additional intelligence for the optimisations can be placed.

- At the original server (in most cases a Web Server)
- At the WAP gateway
- Additional component: a proxy

Figure 4.3 shows the standard client-server concept for a mobile client where an additional module is placed at the original server to perform the optimisation methods.

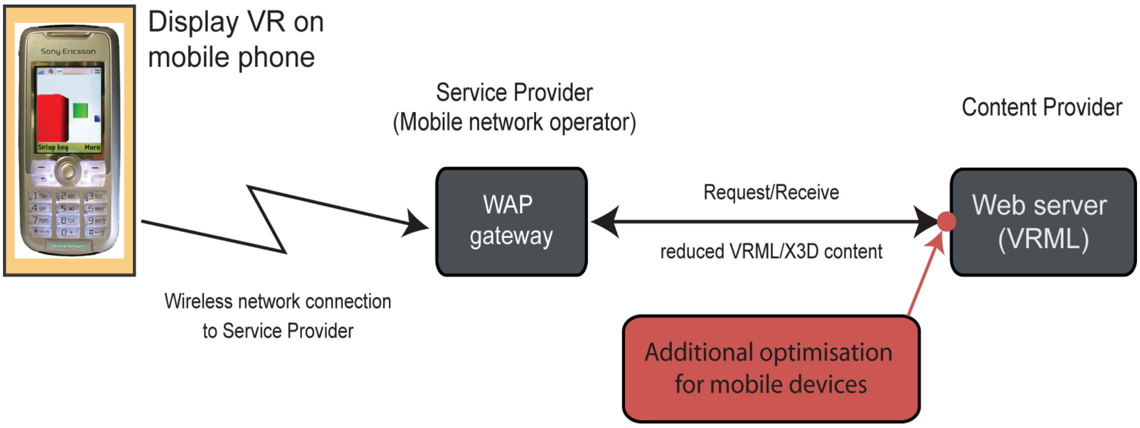


Figure 4.3: Client-server concept with optimisation process at Web server

A disadvantage of this extended concept is that the Web server has to offer the possibility of adding additional modules for the optimisation methods. Another disadvantage is that the client cannot rely on the existence of such modules. The client has to detect the existence of such modules before using them.

Figure 4.4 shows the standard client-server concept for a mobile client where an additional module is placed at the WAP gateway to perform the optimisation methods.

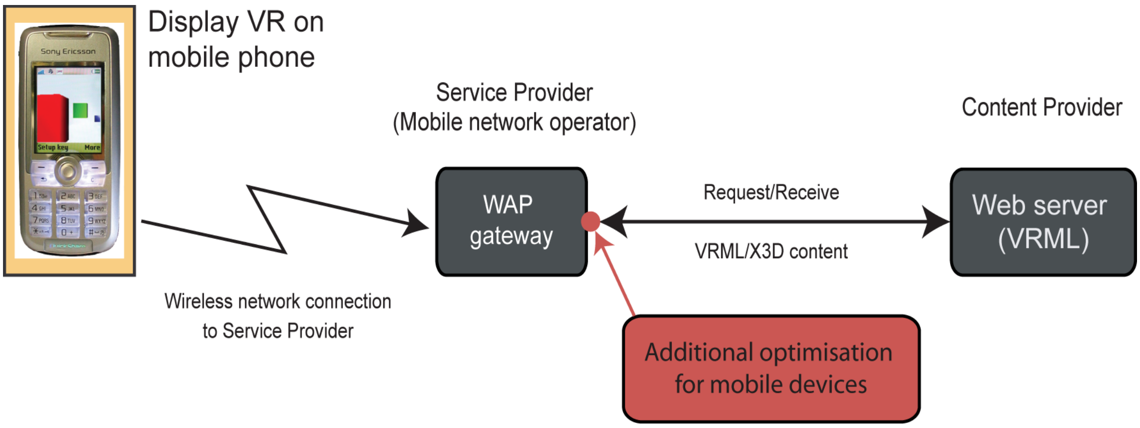


Figure 4.4: Client-server concept with optimisation process at WAP server

An advantage of this approach is that the module will be available for all users which are connected to the Internet over this WAP gateway. Nevertheless, the WAP gateway has to offer the possibility to add such an optimisation module. This is the case if the software is available in source code (open source, kanel) or if the gateway software has a plug-in concept which allows for additional modules. Nevertheless, the client software cannot rely on the existence of these methods because user could reconfigure the mobile device to use another WAP gateway where no optimisation module is available.

Therefore, the client has to check for the existence of optimisation modules at the specific WAP gateway.

The concept shown in Figure 4.5 is an extension of the standard client-server concept. This concept introduces a new component called a proxy. There exists two different ways to use such a proxy, transparent or non-transparent. In the case of a non-transparent proxy, the user has to configure the use of this proxy. This is in contrast to the transparent proxy concept, where the user does not recognise the existence of the proxy.

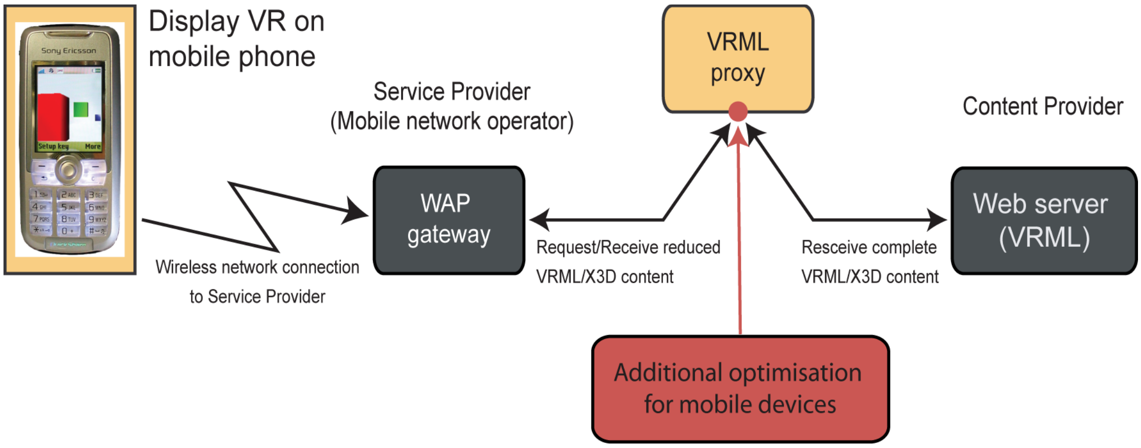


Figure 4.5: Client-server concept with proxy for optimisation

A disadvantage of this concept is that extra delay time is added to the overall delay time between the client and the server. In order to reduce this extra delay time, the proxy component has to be as close as possible to the WAP gateway concerning the network location.

The advantage is that the optimisation modules are always available, no matter which server or client are involved in the connection.

5 Data Transmission and Optimisation: The Proxy Component

5.1 Architecture Overview

The task of the VRML proxy is to filter the content of a requested VRML (X3D) file in order to optimise the data for the client. The VRML proxy is in between the connection of the WAP gateway and the Web server (Figure 5.1). The optimisation can address two different aspects. The first aspect is the reduction of data to reduce the payload over the wireless network. The second aspect addresses the speed up of displaying the received data. This can be achieved by displaying partly received data using streaming technology. Both aspects could be partly counterproductive because the use of streaming technology causes an overhead of data. Nevertheless, the user will not be aware of the VRML proxy, it is a hidden functionality. The goal of the filter process is to optimise the presentation of the Virtual Reality data for the user.

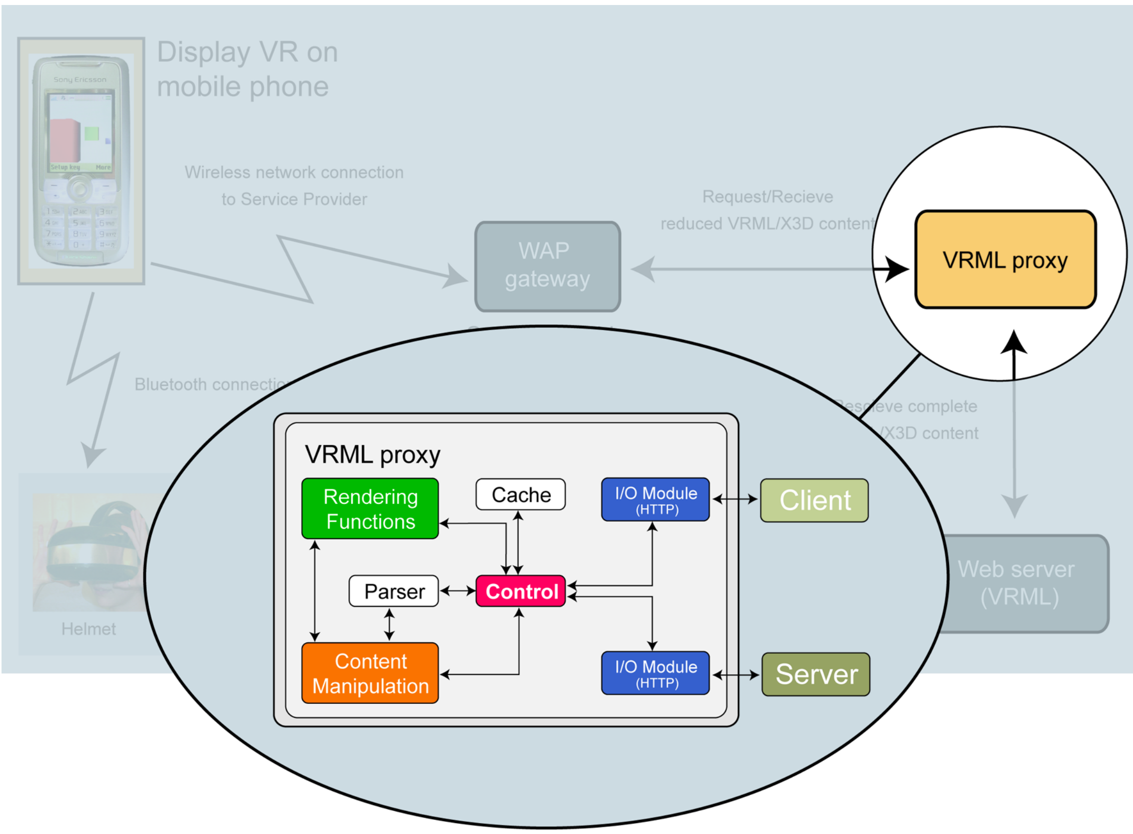


Figure 5.1: System overview with focus on VRML proxy

Additionally, the VRML proxy implements preload and caching methods in order to minimise delay time between client and server where the original file is requested. The caching implementation makes use of special features of the used programming language (Java) in order to optimise the caching process. As discussed previously another method to reduce the amount of data transmission is the compression of data. The VRML proxy makes use of the GZIP algorithm to compress the data.

However, data will not always be compressed. An exception will be done if the amount of the compressed data is bigger than the amount of data of the non-compressed data [EJCIO].

An additional feature of the VRML proxy is the conversion of data into another format. For example this feature is used to converts a picture used as texture from JPEG file format to PNG file format. This feature is more powerful than only converting one image format into another. Additionally it performs a size conversion according the mobile device screen dimensions. The following additional conversions can be performed [HCMC]:

- Any image (JPEG, GIF) into PNG format
- VRML into M3G format
- VRML scene into image (PNG-format)
- Sound (MP3, WMA) into AMR

This conversion features are counter productive to the data reduction aspect and have to used with care.

To get an overview of all tasks of the VRML proxy, Table 5.1 shows which tasks are implemented and where. In the following chapters will discuss the assumption and the resulting consequences concerning the software implementations for the different cases. In Subchapter 5.5 a detailed description of the existing modules is given (Table 5.1).

Task	***Implemented in module(s)***	***Chapter***
Control the application	Control Module	Chapter 5.5.1
Data compression	I/O Module	Chapter 5.5.2
Content data reduction	Parser/Content Manipulation Module	Chapter 5.5.3/5.5.4
Streaming	Content Manipulation Module	Chapter 5.5.4
Data conversion	Rendering Functions Module	Chapter 5.5.5
Cache	Cache Module	Chapter 5.5.6
Preload	Preload Module	Chapter 5.5.7

Table 5.1: List of tasks on VRML proxy and corresponding module

5.2 Data Compression and Decompression

The compression and decompression of data is realised with the GZIP method (LZ77 algorithm and Huffman coding). A disadvantage of this method is that it cannot compress multiple files into one archive [GZIP]. However, this is not a limitation in this project because:

- The GZIP compression and decompression are applied to complete streams (not for archives)
- The application handles a complete input and output stream (request /response)

In order to decompress a request the following procedure is performed:

1. Check header if stream is compressed (content type has to be of mime type “application/x-gzip”)
2. Retrieve the body of the request (as binary data)
3. Construct a GZIPInputStream to decompress the data

In order to compress a response the following procedure is performed:

1. Construct a ByteArrayOutputStream as container of the data
2. Construct a GZIPOutputStream as interface to the ByteArrayOutputStream to compress the data
3. Write bytes to ByteArrayOutputStream over GZIPOutputStream
4. Use compressed data source if smaller, use uncompressed data if not
5. Set header according data source

This compression and decompression methods are used between the connection of the client and the VRML proxy. Figure 5.2 illustrates the way how the output stream compression is realised. The data are compressed and written into a temporarily ByteArrayOutputStream instead of using the output stream to the client.

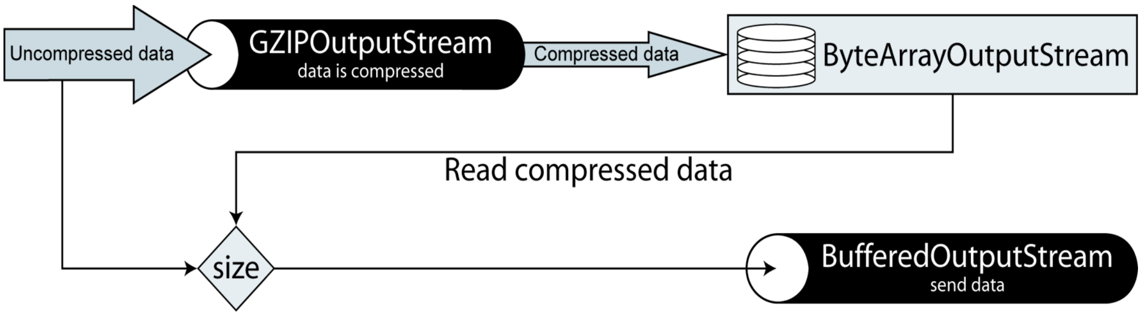


Figure 5.2: Sequence of data output compression for Java

Standard streaming application (e.g. Internet based) can establish a permanent connection. In J2ME such type of connections are jet not available. Thus, the communication between the client and the proxy is HTTP-based [JN, JMR]. Thus every data exchange between the client and the proxy has to be performed as HTTP-based request. In this way the size of the compressed data can be checked and the header value which has to be set before the data are sent can be adjusted. This check is performed for every new object sent out. Thus the compression can be turned on and off during the communication dependent on the size of the actual data.
The functionality of decompression and compression is implemented in the I/O Module (Chapter 5.5.2) of the VRML proxy.

5.3 Reducing Data for Transmission

Data can be reduced with and without loss of information (see Chapter 4). In order to realise the lossless reduction all default values has to be omitted in the output to the client. Therefore, the VRML library had to be modified to perform this task. The output function of every class was modified in such a way that it checks if the actual output parameter is equal to the default VRML parameter. If this is not the case the parameter key and the value are written to the output otherwise the information is omitted. The list of default VRML values for the different nodes (VRML object) is obtained from web3D Consortium [W3C]. Reducing data with loss of information depend on two aspects:

- Node is supported by the mobile device client (supported by the application)
- Settings of the mobile device client

A list of all VRML nodes and the support state is shown in the appendix. A VRML node not supported by the client can be removed because the client application is not able to present the object.
The other possibility lossy reducing of data from the requested VRML file depends on the client settings. The user can specify the VRML nodes the application has to display. According this settings data can be removed from the VRML scene and the transmission of such data can be omitted.
An exception where unsupported nodes are not automatically removed is if first the conversion feature is available (possible) and second the user enabled this feature.
Figure 5.3 shows an example of the visual presentation on a web browser with the Cortona plug-in [PGPC] compared with the visual presentation of the mobile device client with the settings listed in Table 5.2.

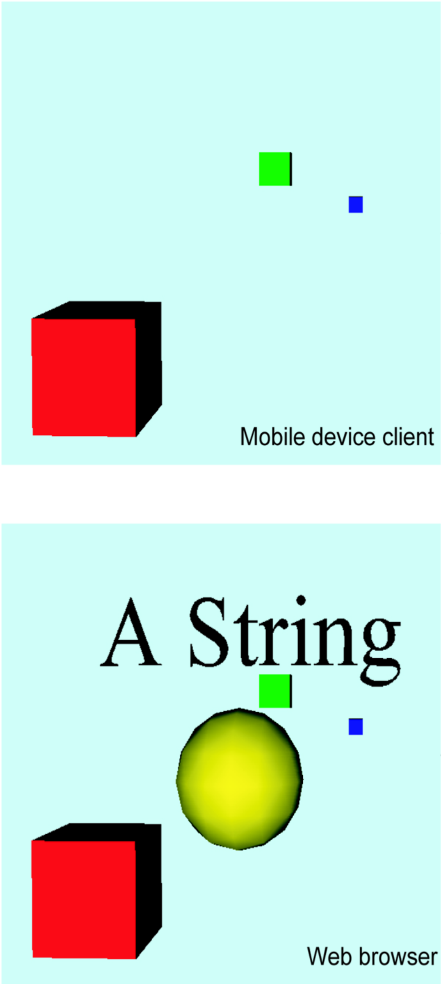
Object	***Texture***	***Dependent of***
Cube	supported	user
Sphere	not supported	user
Audio (wav, midi, etc.)	not supported	client
Text	not supported	client

Table 5.2: Possible mobile device client settings

The data marked red in the VRML source are not displayed on the mobile device because of the settings and thus can be removed by the VRML proxy. There are three different possible cases of the settings such that information is omitted:

- Default VRML value (e.g. box size)
- Not supported by the VRML client application (e.g. text, sound)
- User defined setting (e.g. Sphere)

Visual presentation



Corresponding VRML code

```
#VRML V2.0 utf8

Background {skyColor 0.8 1.0 1.0 }

#Blue Box
Transform {
        children [ Shape { geometry Box { size 2 2 2 }
        appearance Appearance { material Material { diffuseColor 0.0 0.0 1.0 } } } ]
        translation 4 -2 -100 }

#Green Box
Transform {
        children [ Shape { geometry Box { size 1.5 1.5 1.5 }
        appearance Appearance { material Material { diffuseColor 0.0 1.0 0.0 } } } ]
        translation -1 2 -30 }

#Red Box
Transform {
        children [ Shape { geometry Box { size 1 1 1 }
        appearance Appearance { material Material { diffuseColor 1.0 0.0 0.0 } } } ]
        translation 0 0 0 }


#Yellow Sphere
Transform {
        children [ Shape { geometry Sphere { radius 1.5 }
        appearance Appearance { material Material { diffuseColor 1.0 1.0 0.0 } } } ]
        translation 0 0 -10 }

#A Text example
Transform {
        children [ Shape { geometry Text { string "A String" }
        appearance Appearance { material Material { diffuseColor 0.0 0.0 0.0 } } } ]
        translation 0 2 0 }

#A Sound example
Sound {
        direction     1 2 3
        intensity     0.8
        location      1 1 1
        source        AudioClip {
                  description     "A sound example"
                  loop            TRUE
                  url             "examplet.wav" } }
```

Figure 5.3: Source code and corresponding presentation of the VRML data on different clients

5.4 Preparation of the Streaming Data for the Client

VRML (X3D) is an ASCII (UTF8) based description language for 3D Virtual Reality scenes. In the following the term object is used for a graphical object such as a box sphere, etc. while the VRML source code representation is called node [MC97, RCRR97].

The position of the displayed object in the Virtual Reality scene is independent from the position of the described node in the ASCII file.

Figure 5.4 shows an example of the VRML source and the corresponding visual presentation.

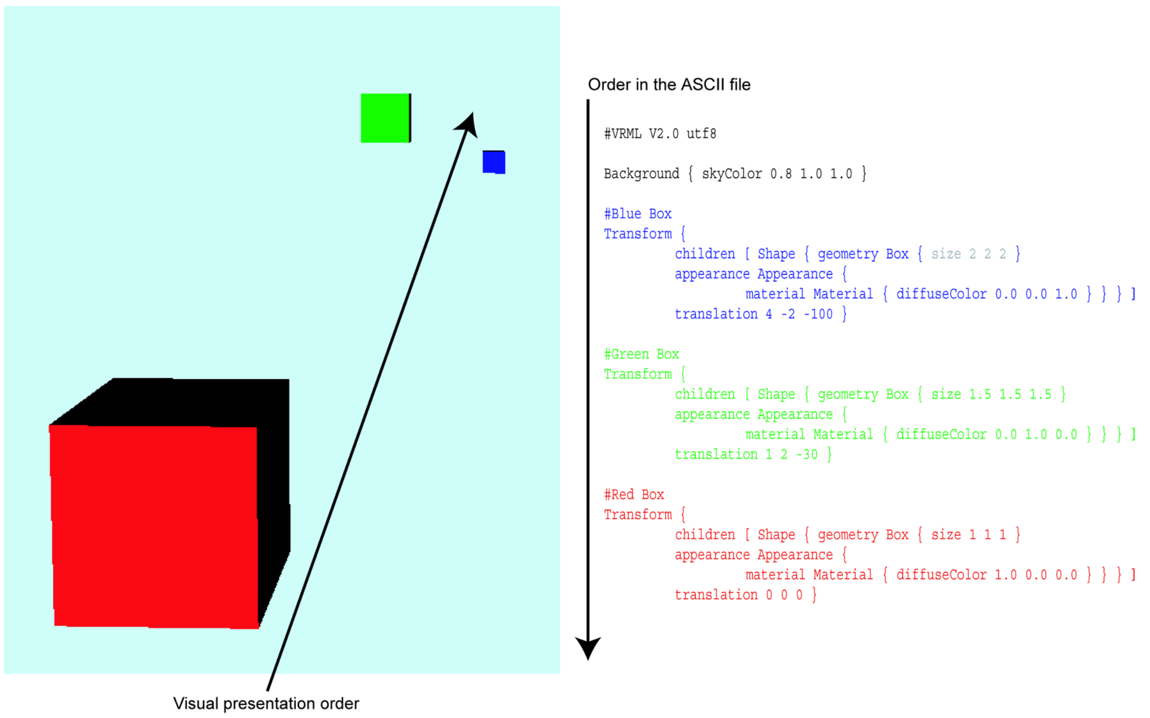


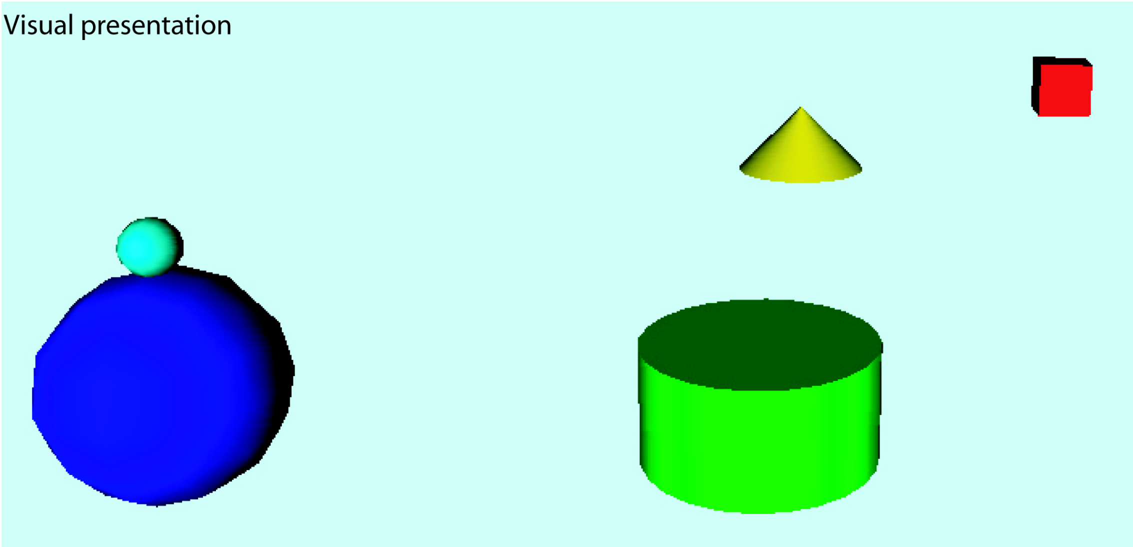
Figure 5.4: Comparison of the VRML file order and visual presentation order

In this example the red box is the biggest and visually most dominant object, followed by the green box. The object with the least visual importance is the blue box. However, in the source file the order of the described object (nodes) is exactly the opposite. In case the VRML proxy uses the order in the file to stream the nodes to the client, the node with the lowest visual importance would be sent first and the node with the highest visual importance would be the last one. In order to obtain a user-friendly behaviour, the node with the most visual presence has to appear first. Therefore, the sequence of the nodes has to be reordered before the nodes are transferred to the client application. The ordering has to be performed according the visual presentation (size on screen). The visual size of an object depends on the following aspects:

- Dimensions of the object
- Position in the scene
- Camera (viewpoint) position

The dimension of the object is specified as an absolute value. It can be easy compared with other objects of the same type. However if an object is of a different type the shape is different as well. Even in case of simple objects such as a box or a sphere the comparison process is complex and with the complexity of the objects the complexity of the comparison process increases.

Figure 5.5 shows the source of a VRML file and the resulting visual presentation of different nodes at different positions in the scene. The highlighted red and orange parts of the source code influence the visual size of the nodes.



Corresponding VRML code

```
#VRML V2.0 utf8

Background {skyColor 0.8 1.0 1.0 }

Transform { ①
        children [
        Transform { ⑤
                children [ Shape { geometry Sphere { radius 1 } appearance Appearance { material Material { diffuseColor 0.0 0.0 1.0 } } } ]
                translation 0 0 0 }

                Transform { ⑥
                children [ Shape { geometry Sphere { radius 0.25 } appearance Appearance { material Material { diffuseColor 0.0 1.0 1.0 } } } ]
                translation 0 1.2 0 } ]
         translation -5 0 0 }

#Green Cylinder
        Transform { ②
        children [ Shape { geometry Cylinder { height 1 radius 1 } appearance Appearance { material Material { diffuseColor 0.0 1.0 0.0 } } } ]
        translation 0 0 0 }

#Red Box
Transform { ③
        children [ Shape { geometry Box { size 1 1 1 } appearance Appearance { material Material { diffuseColor 1.0 0.0 0.0 } } } ]
        translation 5 0 -15 }

#Yellow Cone
Transform { ④
        children [ Shape { geometry Cone { height 1 bottomRadius 1 } appearance Appearance { material Material { diffuseColor 1.0 1.0 0.0 } } } ]
        translation 0 0 -10 }
```

Figure 5.5: VRML code with grouped objects and the corresponding visual presentation

The dimension is not the only aspect which influences the visual presentation of the nodes. A big object far away from the "viewer" will occur smaller compared with the same object close to the viewer. Thus the absolute position of the object in the scene and the camera position result in a relative distance between camera and object. This relative distance together with the object shape specify the visual size of an object.

A simple and reliable way to perform the ordering of the VRML object according to visual size consist in the following: One can use the same rendering functions which are used to display the nodes. Each node is taken separately and rendered in an off-screen buffer. This off-screen buffer is examined according to the pixels different from the background. The more pixels are counted in the off-screen buffer for a node the bigger is the visual presentation.
This method can be tuned to take lightning effects into account. In this case not only the difference of the pixels respect to background is important. Each pixel different from the background is additionally weighted to the original colour. This method can even be improved by taking the contrast into account.

If a light yellow object is compared with a black object of the same size and relative difference to the camera there is no difference in the visual size. However, on a white background the black object will have a bigger visual dominance. A series of small not representative tests have shown to have low influence on the ordering. Therefore it was not implemented in the VRML proxy.
An important question is the following: Which nodes or group of nodes have to be considered for the ordering. Figure 5.6 shows the nodes of the previous example in a hierarchical structure with all children and subchildren [MC97, RCRR97].

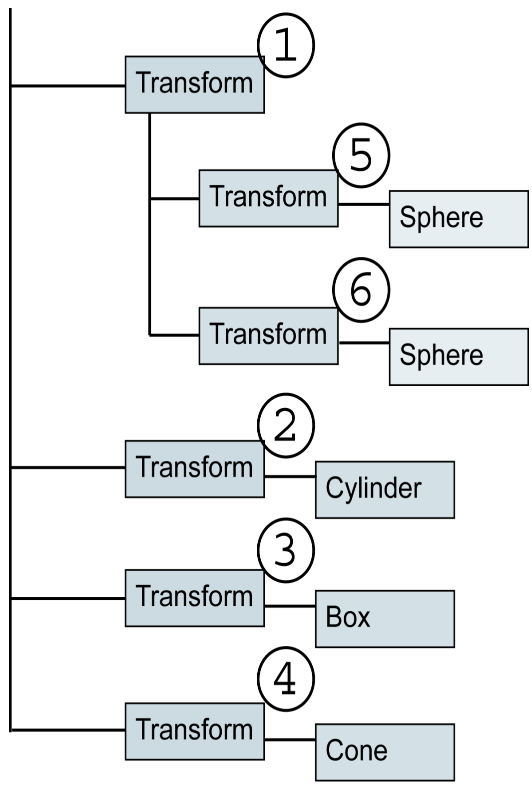


Figure 5.6: Hierarchical VRML object structure of Figure 5.5

In the previous example there are five objects visible but the scene consists out of eleven nodes. The nodes 2,3 and 4 are directly connected with a shape node (cylinder, box and cone). Since the nodes 2,3 and 4 have only one child one can directly take this node (including the child) in to consideration.

The nodes 1, 5 and 6 are more difficult to handle: The node 1 has more than one child. The node children 5 and 6 are directly connected with a shape (sphere). At this point two possible decisions can be made:

- Directly use node 1 with the subchildren 5 and 6
- Consider two cases:
  1. Node 1 with subchild 5
  2. Node 1 with subchild 6

In the first case both visible nodes (both spheres) are used for the ordering decision. In the second case the decision for the ordering is performed for the spheres separately. In order to decide which of the two cases has to be used, different aspects has to be considered and compared.

Use node with all subchildren (group):

- Several small objects becomes one big
- Grouping remains
- Data are not increased

Use node with one subchild:

- Small objects stay small
- Grouping destroyed
- Data are increased (Transformation for a group has to be added to every object)

A disadvantages of taking every node separately into account is that the amount of nodes will increase and thus the amount of data. If nodes are regionally close together (spheres in Figure 5.5) they can also be considered as one object. In such a case it will not be an disadvantages to use the whole group of nodes for the comparison process. This is different in the case the grouped nodes are far away from each other. In most of the cases VRML files are created by humans. Thus, grouping nodes is used because the nodes belong together usually and are close together regionally. As result this point will be not considered as disadvantage.
As consequence the method where the group of nodes is taken into account for the comparison process is used.

5.5 Module Description

In order to perform the tasks described in chapter 4 a VRML proxy with the structure illustrated in Figure 5.7 was developed. The tasks the proxy has to fulfil are given in Table 5.1.

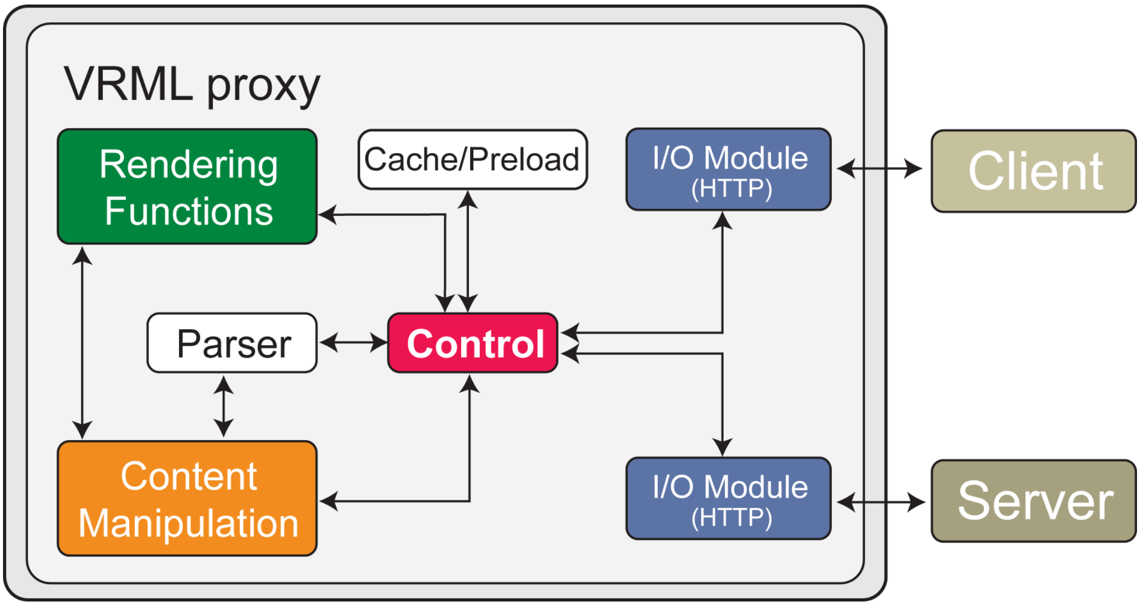


Figure 5.7: VRML proxy system overview, showing the realised software components to perform the optimisation

Figure 5.8 illustrates a typical communication sequence between the client and the VRML proxy and the involved modules.

The client request received by the VRML proxy is evaluated and validated by the I/O module. Additionally, this module will load the requested VRML file from the Web server. After the VRML file is parsed and an internal representation is established, the content manipulation can be performed. The resulting optimised data is sent back to the client.

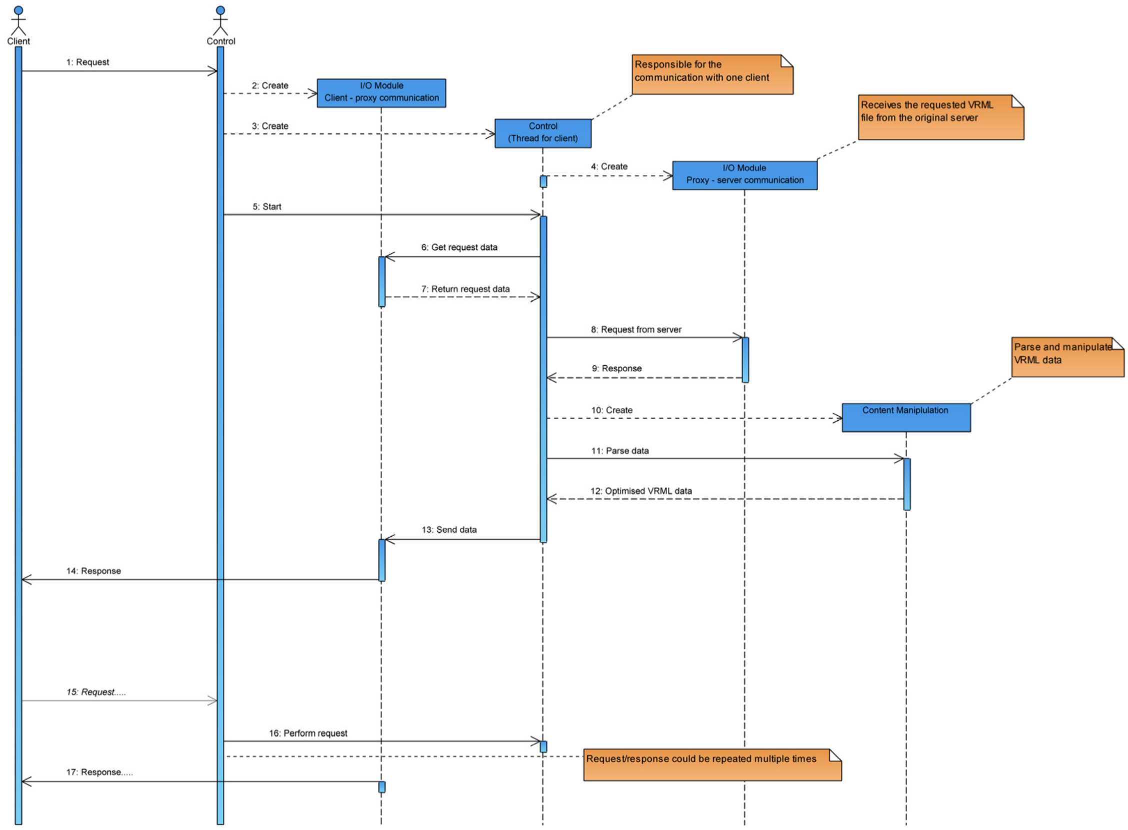


Figure 5.8: General request/response sequence diagram

5.5.1 Control Module

The Control module is the core module of the system. Figure shows the sequence diagram of a standard request.

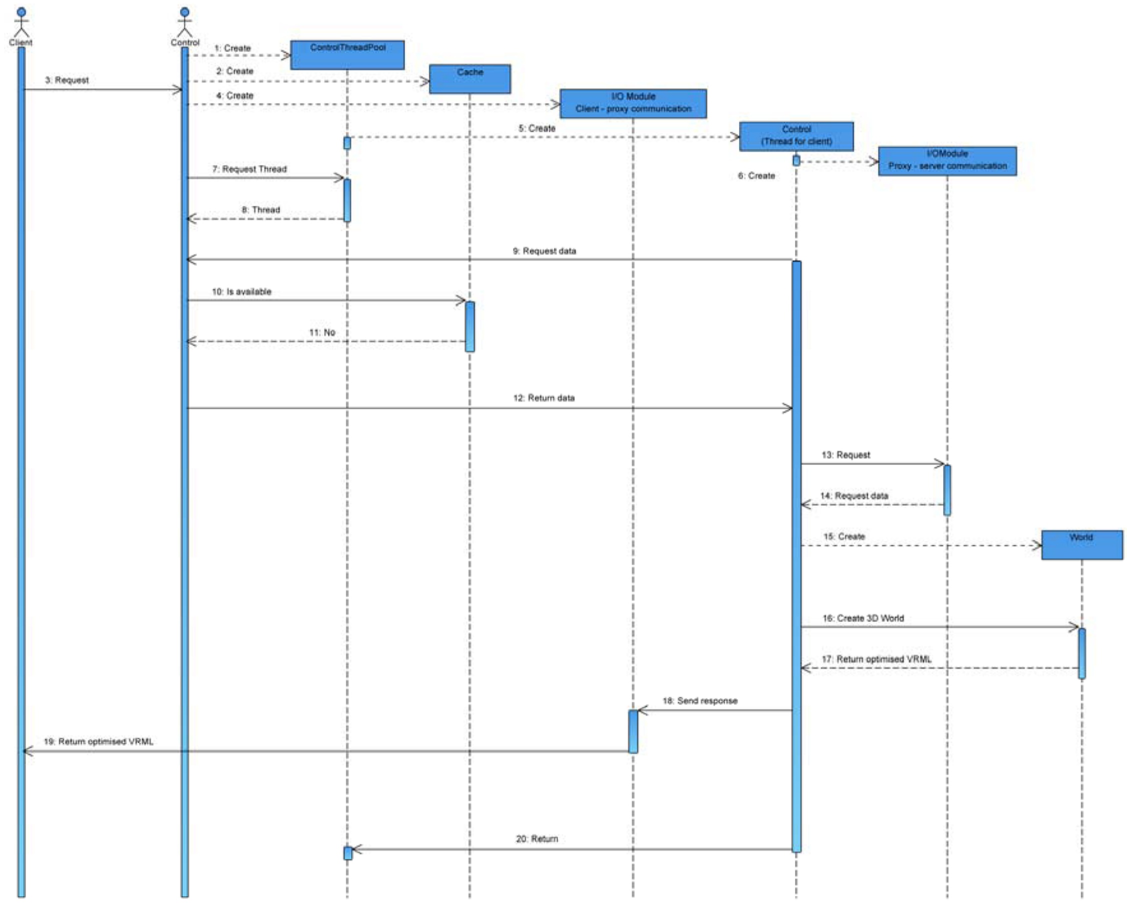


Figure 5.9: Sequence diagram for a standard request

The task of the single main Control module is to listen to a port. After the main Control module has received a client request, it will construct an I/O module (see 5.5.2) which will provide the information of the request [EJCT]. The main Control module extracts the session ID and based on this information performs the following decision:

- Request a new Control module which will be linked with a specific client
- Forward the request to the Control module already linked with this child

The first case holds if the request contains no or an empty session ID. In the response the VRML proxy appends a unique session ID used by the client within every further request. A specific user is identified according to the session ID. This ID has to be submitted within every request by the client. Therefore the Control module can perform the association between a specific client and the specific Control module.

There is no other way to handle that association between the client and the Control module because the Java API on a mobile device does not offer the possibility to keep a connection alive. However, this way of communication eliminates the need of maintaining a constant connection and handle connection abortions.

The association between a client and the corresponding child process is important because only this child process has loaded the scene and the mobile device configuration data. It can execute the requested functionality with a minimum of information.

The associated thread Control module is responsible for all requests coming from the linked client to make the following decisions:

- Request a new VRML (X3D) file from a server
- Use a cached version of the file
- Perform manipulation on the already loaded VRML scene

5.5.2 I/O Module

The I/O module is responsible for both communication between the client and the VRML proxy and the communication between the VRML proxy and the server providing the original 3D Virtual Reality data file. Since these files are located on Web servers the communication is an HTTP-based client-server communication, which allows the use of a third-party library [APACHE]. The I/O module uses the basic communication functionalities of this library and wraps the logic around. Figure 5.10 shows the sequence diagram of a standard I/O communication.

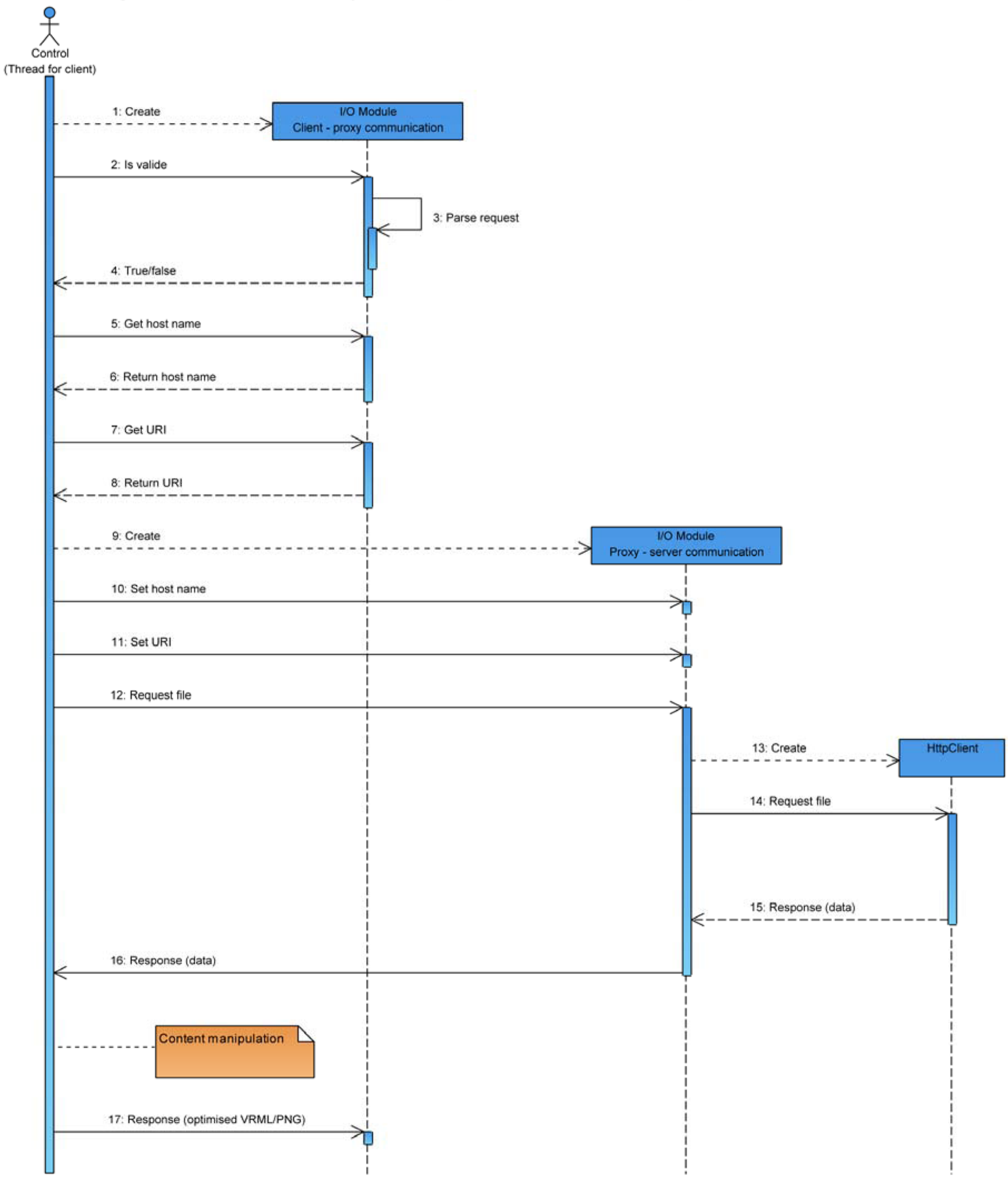


Figure 5.10: Sequence diagram for the I/O component

As illustrated in Figure 5.10, the I/O module can have two different functions. First the I/O module can be responsible for both communication with the client and second for the communication with the Web server. Therefore Figure 5.10 shows two I/O modules which are two different software instances of the same I/O module (class). In case of "client", the parsing of the HTTP request from the mobile client is because of the simplicity done by an own written method. This method provides the following information:

- Checks if request is valid (true/false)
- Retrieves the information of the host
- Retrieves the information for the file
- Retrieves the session ID

The response to the mobile client is compressed if the data amount is smaller otherwise the data is send in uncompressed way. Additionally, the following header information is set:

- Content type: text/plain, model/vrml, image/png or application/gzip
- Session ID

In case of the third-party library for the HTTP-based communication is used to request the file from the Web server. The main reason for the use of the third-party library is the high complexity of the communication. For example, a Web server often does a "chunked" output splitting the response in multiple packages [RFC2616]. The response from the Web server is obtained in bytes (ByteArray), thus it is possible to receive ASCII based data (VRML, X3D) as well as binary data (JPEG, PNG).

5.5.3 Parser Module

The Parser module receives the loaded file from the server and assembles a software object representation of the 3D Virtual Reality scene. The internal 3D data is represented in Java3D [J3ASE]. After the object representation is established the control is delegated to the Content Manipulation module.

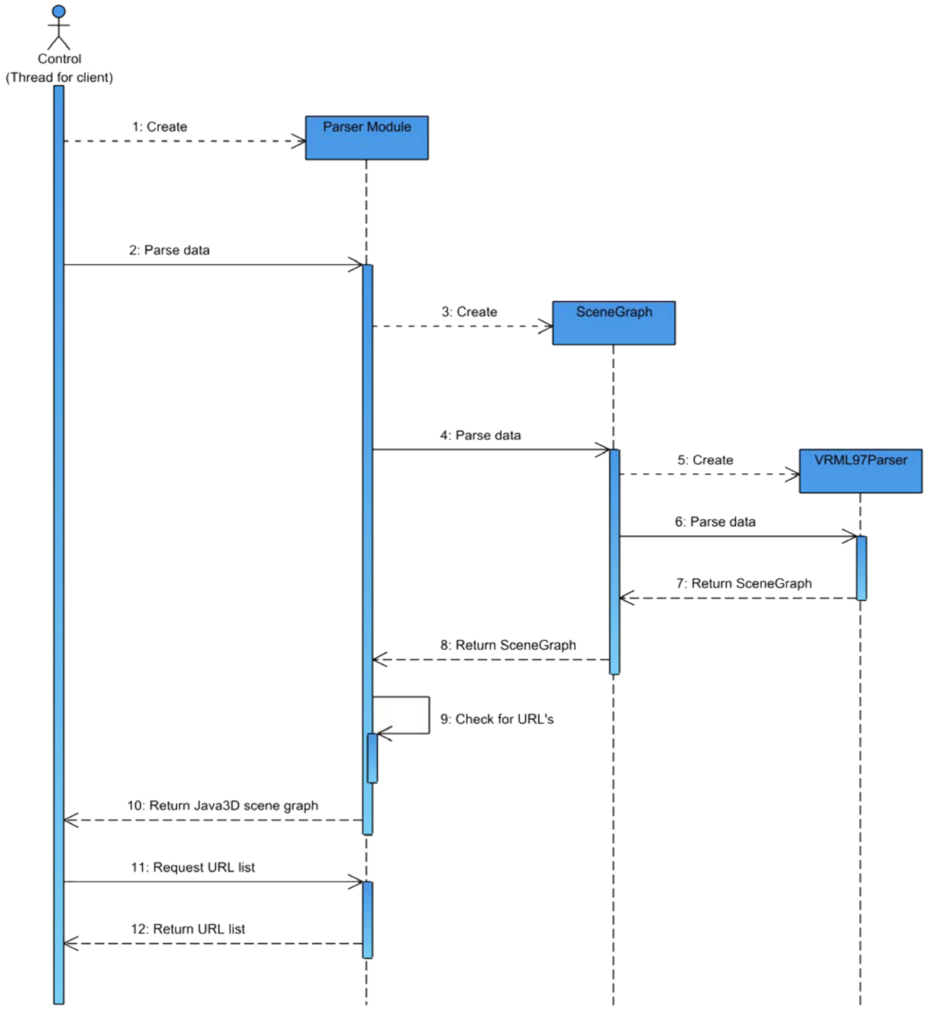


Figure 5.11: Sequence diagram of parser module

The parser is able to read VRML data and to transform the data into a Java3D representation. For this task a third-party parser from Satoshi Konno is used [VLC]. This parser is able to handle 3DS, DXF, OBJ, NFF, STL, and the newest version [1.0] adds support for X3D. Therefore, it is possible to visualise these formats on the mobile device because the internal Java3D representation is retransformed into VRML and submitted to the mobile device. Parsing the file and establish an internal software object representation (Java3D) makes it possible to manipulation data. With the help of this object representation which reflects the size, distances and shapes of the textual VRML file data manipulation and/or streaming functionality can be performed.

5.5.4 Content Manipulation Module

The Content Manipulation module handles the 3D Virtual Reality scene. It is able to strip down the scene, remove unneeded and unsupported content from the mobile device. To do so, the mobile device sends the device-specific information (screen size, memory, brand and model name). Furthermore, the mobile device transfers user-specific information to this module such as enabling specific support for textures, sound and animation. The mobile device screen size is important for the decision whether a node should be sent. If a node is big enough relatively to the screen it is transferred. Otherwise it is assumed to be too small to appear on the mobile device screen.

This module performs the basic tasks needed for the “streaming”. It performs the ordering of the objects according to the user position, size and structure of the objects and the relative object size. If desired it removes additional information such as textures, sound and animation. In addition, it performs conversion of textures and sound files. Since Java3D only supports textures in the PNG format a conversion from other file formats such as GIF, JPEG, etc. has to be performed [PNG]. The same is valid for sound data. In addition, these conversions can be used to scale the images down and thus saving transmission time and bandwidth and reducing costs.

Additionally, the user’s position is updated. Therefore the loaded scene directly reflects the scene at the client side. According to this information the objects relative size are rechecked and adjusted. Therefore objects in background which were rejected because of the relative size can become of importance if the user goes through the scene. If this case occurs, the information is transferred to the client on the mobile device.

To obtain good quality for personal computer based viewing the texture files have to be high resolution images which are in most cases large in file size. Because of the screen size of the mobile device this high texture quality is not needed. In order to further optimise the data resolution conversion will be performed during format conversion.

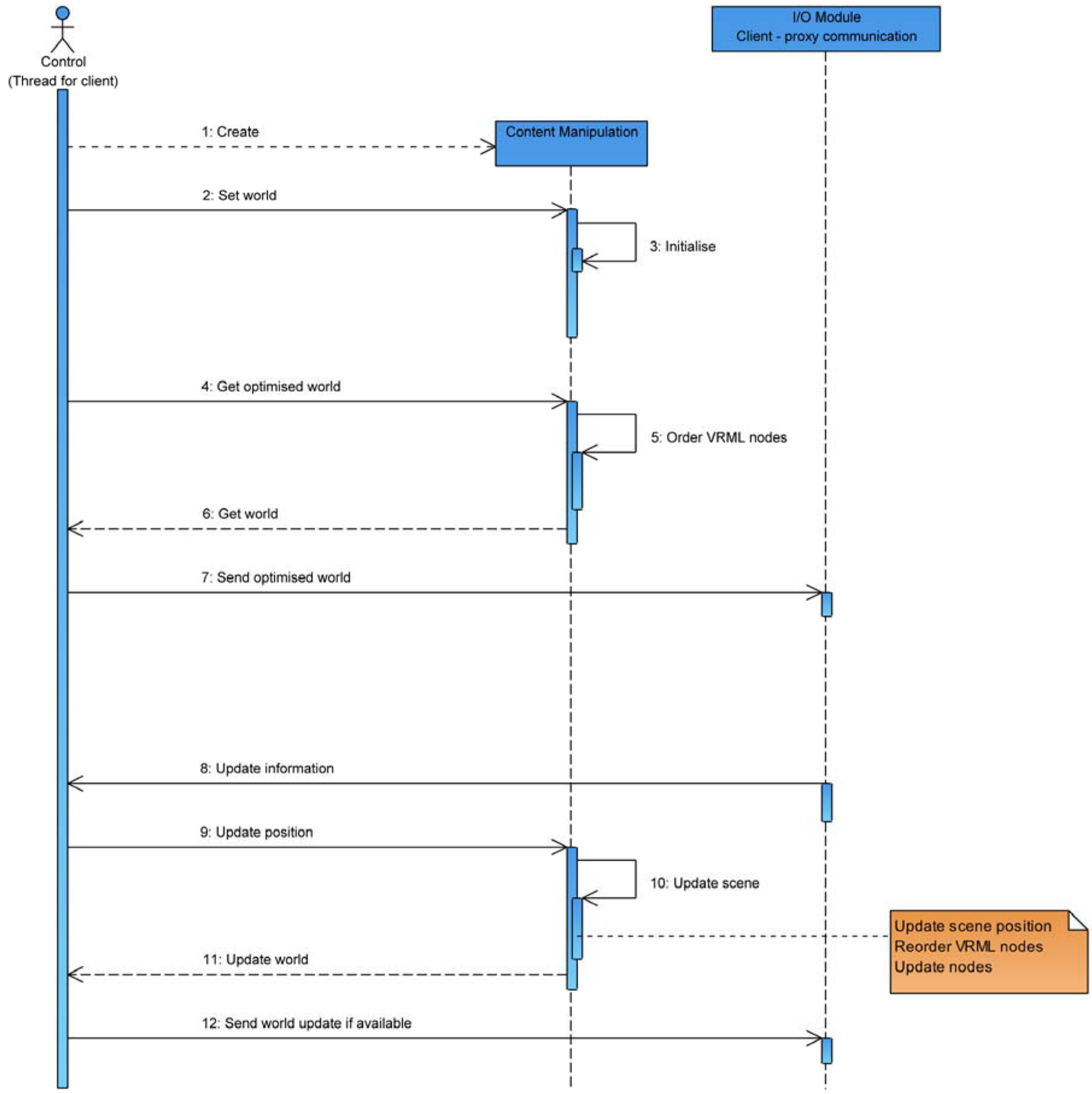


Figure 5.12: Sequence diagram of Content Manipulation module

5.5.5 Rendering Functions Module

The Rendering Functions module provides remote functions for the mobile device. Communication with the Content Manipulation module allows this module to directly use the local scene to work and process on. Thus, the communication between client and VRML proxy is reduced to a minimum.

This module is not useful in actual GSM, HSCSD and GPRS networks because of the high latency time. The required time to perform a time consuming action is comparable to the time necessary to send information to the VRML proxy, execute the function and send the response. The more processing power the mobile device has the worse this situation becomes.

The reason for keeping this module in the system is that the situation in Third Generation mobile networks (UMTS) is different. An additional reason is that in future J2ME versions of the Remote Methods Invocation (RMI) feature will be implemented [MRMI, SYM04]. This feature simplifies the execution of remote functions on the VRML proxy.

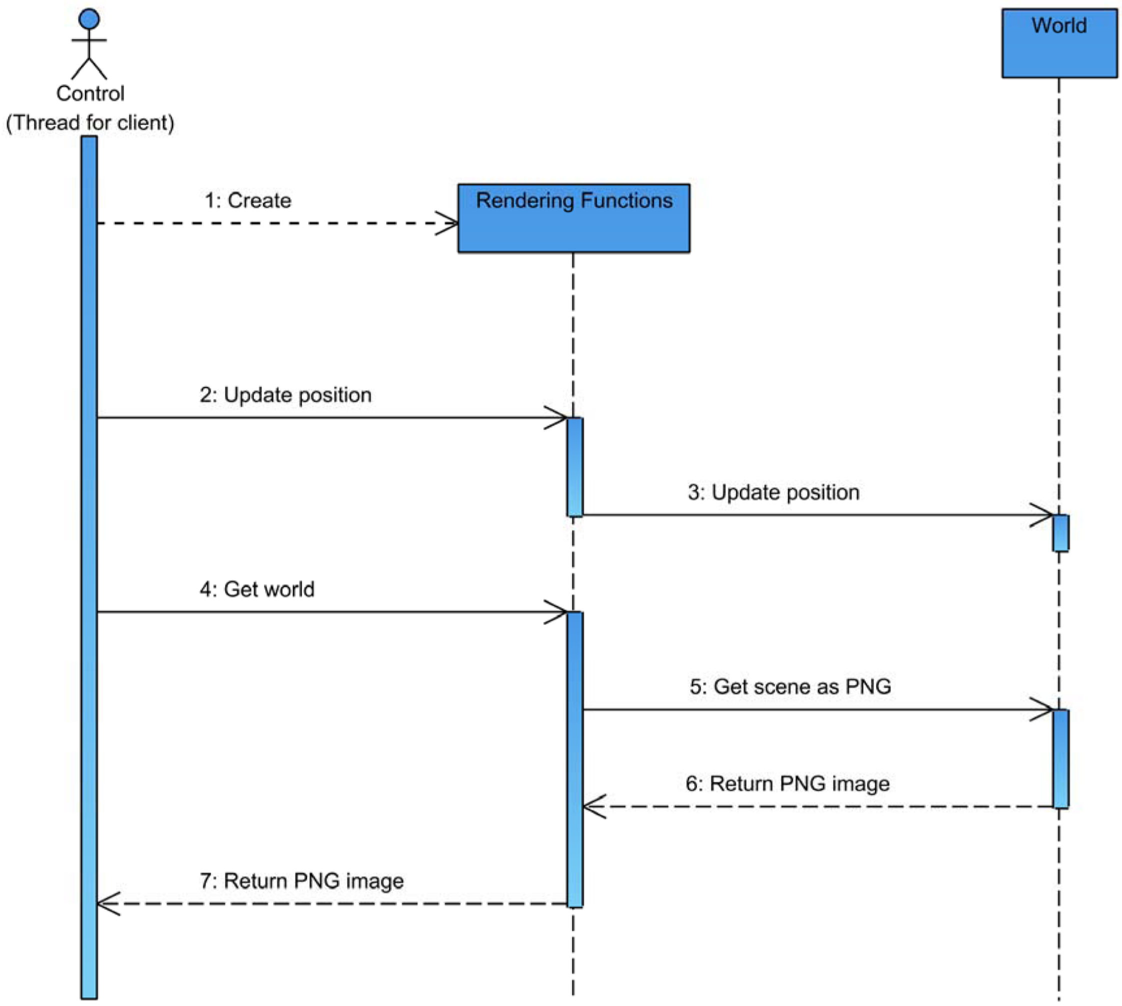


Figure 5.13: Sequence diagram of Rendering Functions module

5.5.6 Cache Module

The Cache module stores and manages the cached versions of the 3D Virtual Reality data. Instead of saving the original file received from the server the module saves the parsed object-oriented representation. To perform this task the Java feature called “serializability” is used [EJCIO]. This features allows to save a software object representation established in the main memory into a file on the hard drive.

The advantage is that the file has not to be parsed in order to establish a software object representation. When reading out the cache all software objects with all links are directly reconstructed in the main memory and thus the object representation can be used directly. This feature increases the performance of the cache.

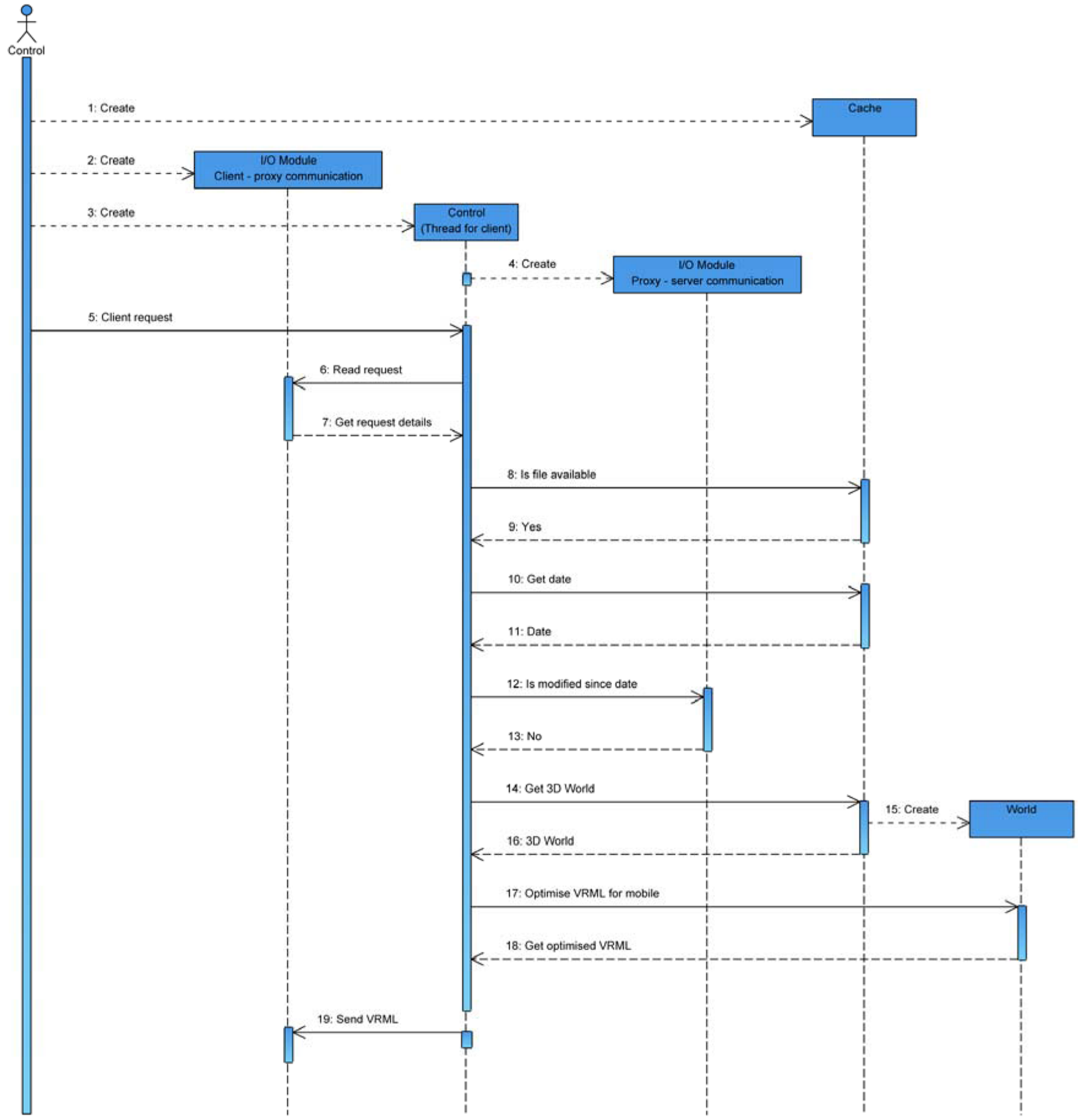


Figure 5.14: Sequence diagram of Cache module

5.5.7 Preload

The Preload module addresses the optimisation of transmission between the VRML proxy and the Web server. Figure 5.15 shows the sequence diagram of the preloading functionality. With the help of the Parser module a list of URL's found in the loaded VRML file are obtained. This URL's are linking textures, sounds or other VRML scenes with the actually loaded one. While the content manipulation on the actual VRML scene is performed the data of the URL's are loaded. In case the client requests any of these information the data are already loaded and the delay time is reduced.

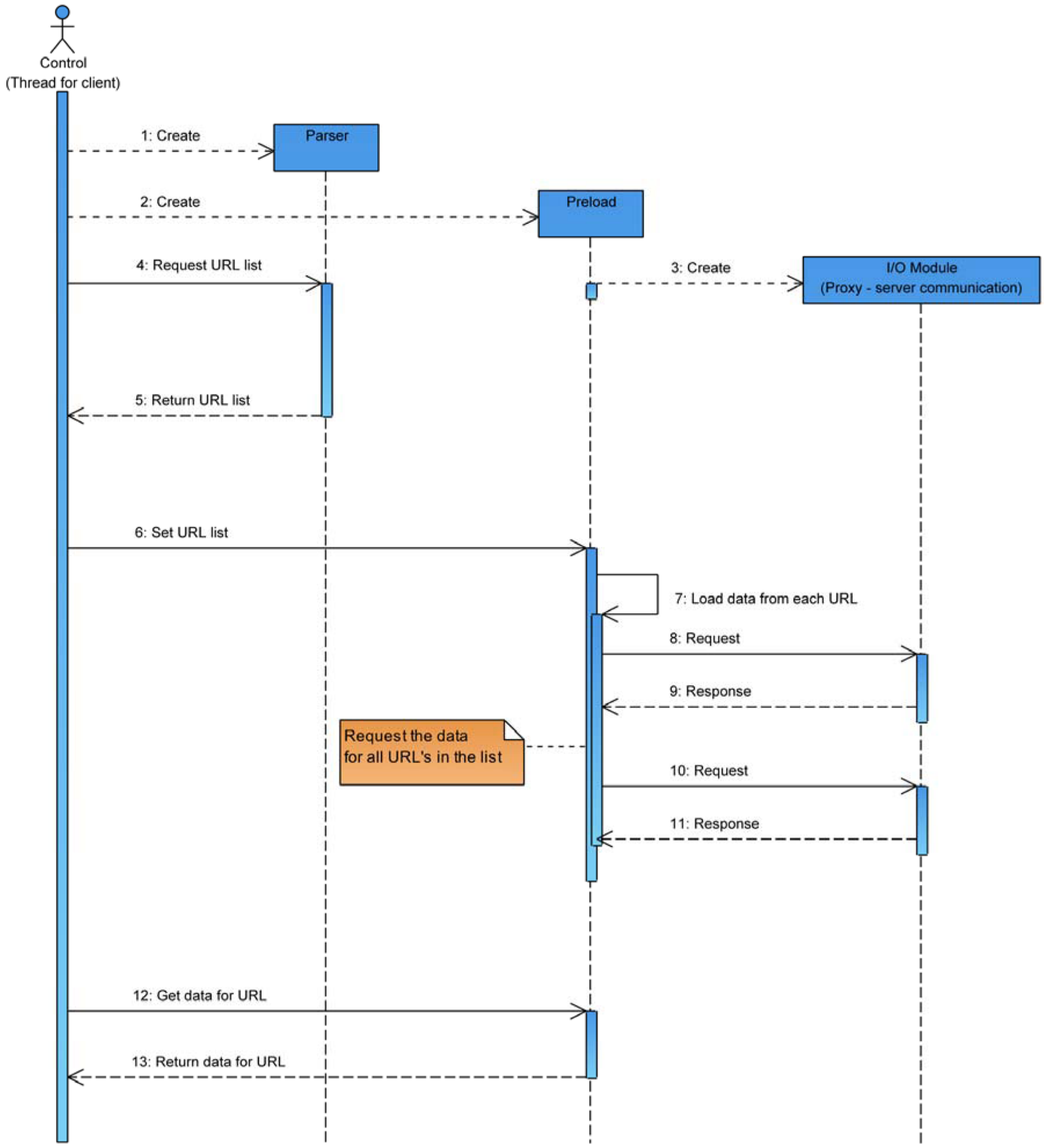


Figure 5.15: Sequence diagram of preload

6 Display the 3D Data

6.1 Generation and Visualisation of 3D Data

6.1.1 General Introduction

There are different projection methods which are traditionally used by painters and technical drawers to give people a realistic image. These techniques can also be used to draw a 3D image on a 2D screen (monitor).

The following projections are known as planar geometric projection [FDFH94]. The planar geometric projection itself are divided into two basic classes: The perspective projection and the parallel projection. Figure 6.1 shows the subdivisions of the planar geometric projection.

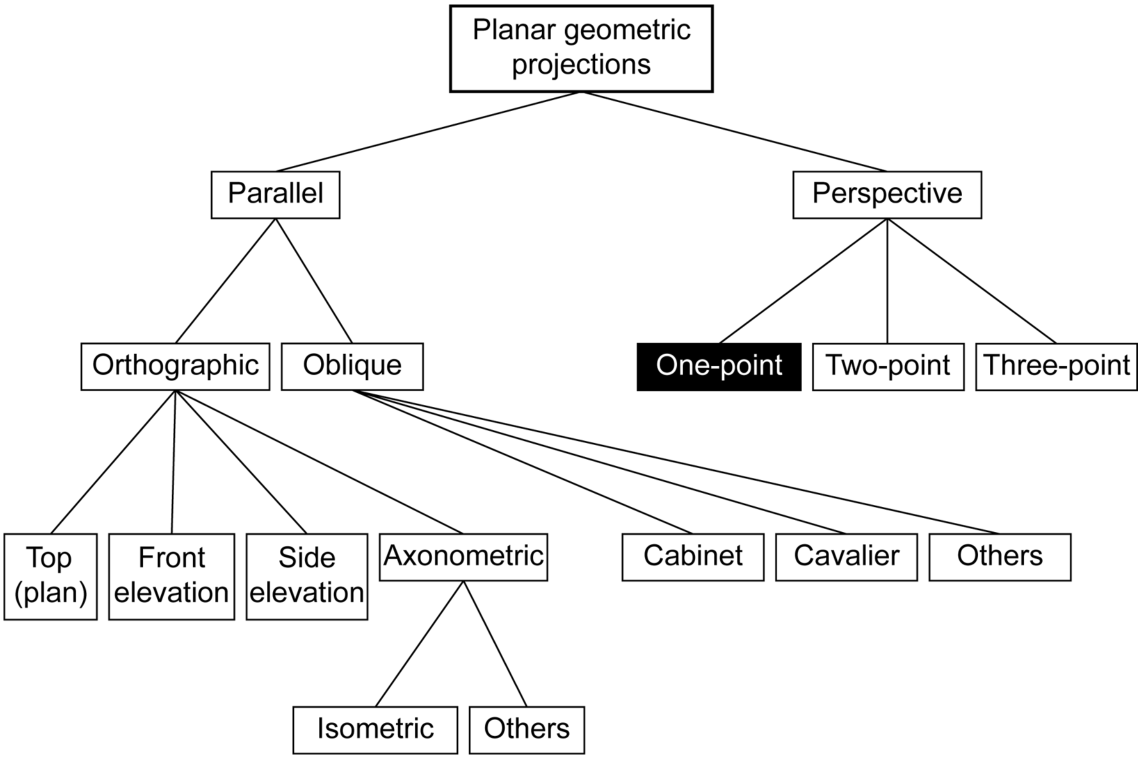


Figure 6.1: Hierarchy of the different projection methods

The perspective projection shown in Figure 6.2 is similar to the human visual system. In contrast to the parallel projection where the centre of projection is at infinity, therefore the projectors are parallel. The advantage of the parallel projection shown in Figure 6.3 is that parallel lines remain parallel, so this projection can be used for exact measurement. This is not true for the perspective projection.

Although the perspective projection has this disadvantage it is used to display 3D (in Java3D, VRML, etc.).

The reason for this fact is that the perspective projection is more realistic than the parallel projection because it is similar to the human visual system.

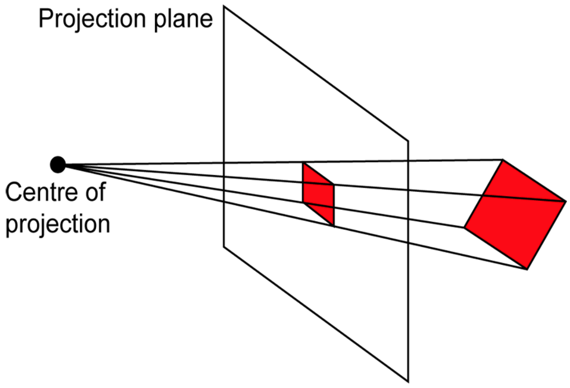


Figure 6.2: Perspective projection

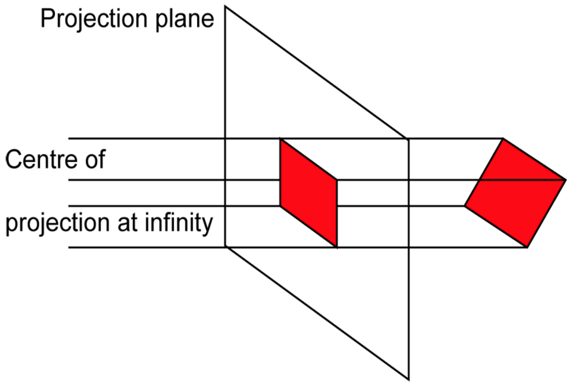


Figure 6.3: Parallel projection

Figure 6.4 shows the principle of the method. The 3D object, in this case a simple cube, should be displayed as a 2D image. In the initialisation step, the virtual 3D area, the image plane and the centre of projection (often denoted as eye or camera position) as well as the distances between them have to be specified. This information is needed for the calculation of the 2D image. The next step is to insert the cube in to the virtual 3D area.

The projectors are calculated from the centre of projection to the cube. The edges of the cube are at the position where the projectors cross the projection plane. For example point A and C correspond to the points A´ and C´ on the projection plane.

The projection plane is parallel to the v/u plane and therefore crosses the n-axis perpendicular. To avoid confusion with the “real world coordinate system” these axes are typically called the v, u and n-axis.

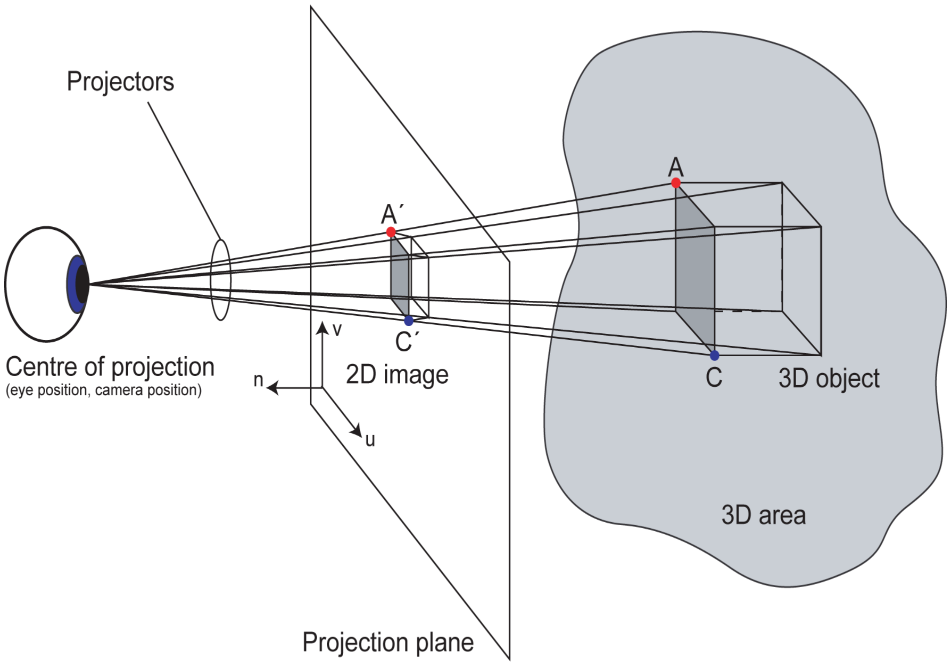


Figure 6.4: One point perspective projection of a cube

To improve the realism of the obtained images as result of a one point projection, a two point projection or even a three point projection can be used. The improvement of the realism between a two point projection and a three point projection is not big, therefore it is not often used. The two point projection (Figure 6.5) is used in science, architecture, industrial design and advertisement [FDFH94]. Figure 6.5 shows a two point projection where all lines which are parallel to the v-axis (normally this axis is used in a right hand system) stay parallel. For the two other axes, a vanishing point is introduced. All lines which are parallel to these axes end in the vanishing point. For the three point projection, all three axes end in a vanishing point.

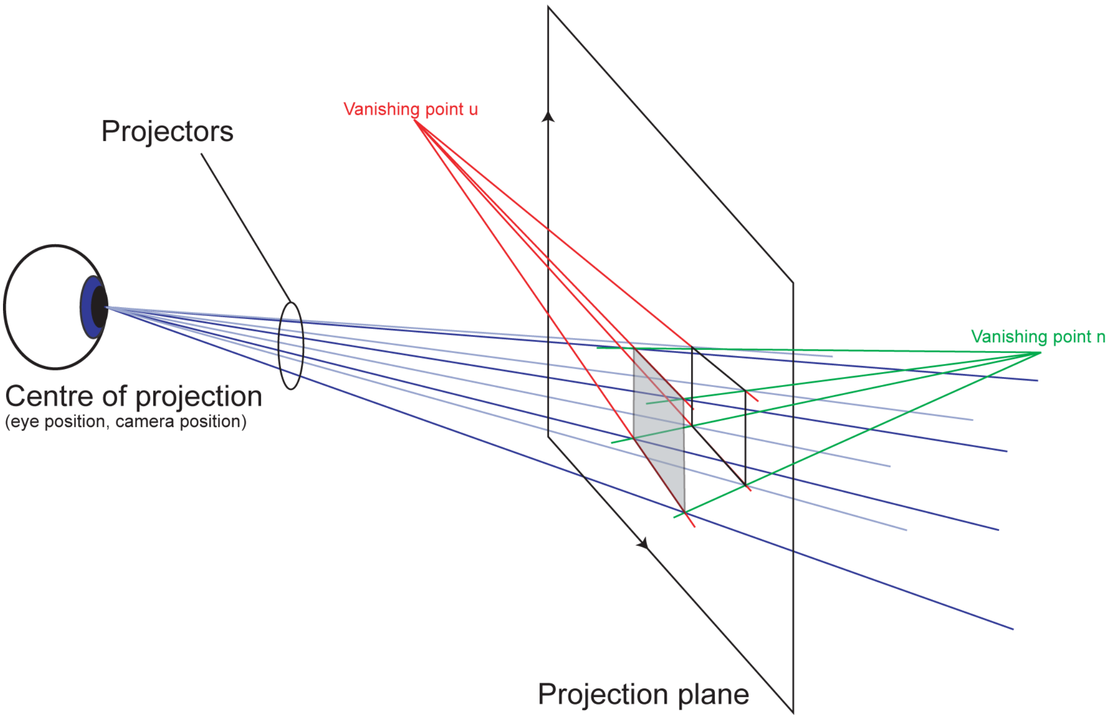


Figure 6.5: Two point projection with the two vanishing points u and n

By applying one these projection methods (Figure 6.1) the problem of displaying a 3D world on a 2D screen can be solved. Software implementations of these algorithms are available as libraries (API), as commercial product and/or as freeware. Available libraries are [HCMC, WMDD, J3ASE]:

- Direct3D Mobile (Windows)
- Dorado (Symbian, Windows CE, C/C++ API, JNI support planed)
- HI Corporation's Mascot Capsule (Symbian, C/C++ and/or Java)
- Java3D OpenGL/DirectX

These libraries use proprietary or open standards [JCP, WMDD]:

- DirectX
- OpenGL
- JSR 184 (M3G) Mobile 3D Graphics API for J2ME
- JSR 189, Java 3D API
- JSR 239, Java Bindings for OpenGL ES (Java, early drafts, influence on JSR 184)

6.1.2 Generation of 3D Data Presentations (Java3D, M3G & VRML)

This research project is based on Java 3D, M3G & VRML. Therefore algorithms for generating 3D data presentation were studied. Since VRML is a file format it does not specify any algorithm for generating the visual two dimensional representation. The same is valid for the specification of Java3D as well as the M3G specification.

Nevertheless, the libraries which implement the specifications have to perform this task. To display a VRML Virtual Reality scene the Cortona browser plug-in and the Cortona PDA client were used. The Java3D library as implementation of JSR-189 and the HI Corporation's Mascot Capsule library for Java as implementation for the JSR-184 were used. It has shown that all libraries and clients are using the same principle to render the Virtual Reality scene to display it on a screen. They are using the previously described one point perspective projection method. However additional aspects have to be considered.

All methods mentioned above of the projected scenes where unlimited in dimension and projection plane. In order to display a 3D Virtual Reality scene, a view volume has to be introduce the 3D Virtual Reality scene is clipped against. With the projection and the view volume all needed information is available to clip and project the 3D Virtual Reality scene on a 2D plane. The following description is obtained from [FDFH90, FDFH94, RAU93].

Figure 6.6 shows how to define the view plane and the viewing reference coordinate system (VRC) with the help of a view reference point and two vectors. One of the two vectors is the normal vector of the view plane (VPN) and the other is the so called view up vector (VUP). The view reference point (VRP) together with the view plain normal vector define the view plane (Hesse-Normal-Form, HNF). The origin of the viewing reference coordinate system is the VRP. The VPN itself defines one of the coordinate axes (n-axis). The projection of the VUP, parallel to VPN onto the view plane specifies the v-axis. The u-axis is defined in such a way that it forms a right hand coordinate system with the n- and the v-axis.

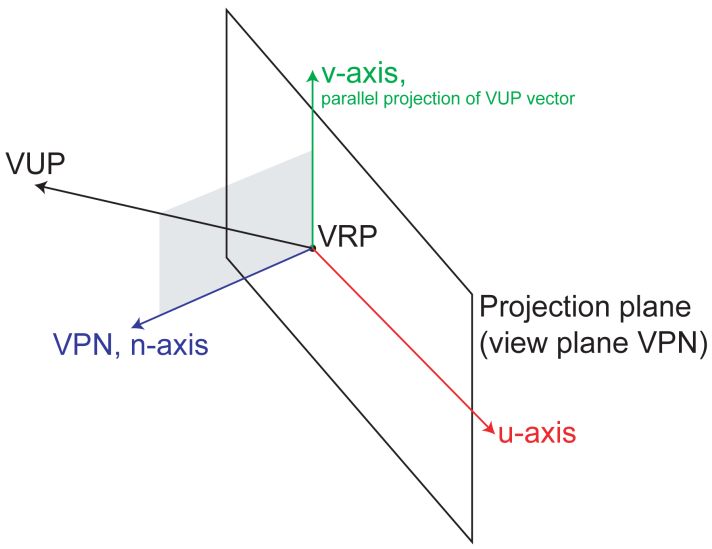


Figure 6.6: Generation of the view plane, viewing reference coordinate system (VRC)

So far the 3D scene was not clipped. For this reason, a window is introduced, which is located in the view plane. Everything outside this window will not be displayed (on the screen).
The centre of this window has not to be equal to the VRP. Figure 6.7 shows the view plane together with an example of the projection centre and the clipping window.

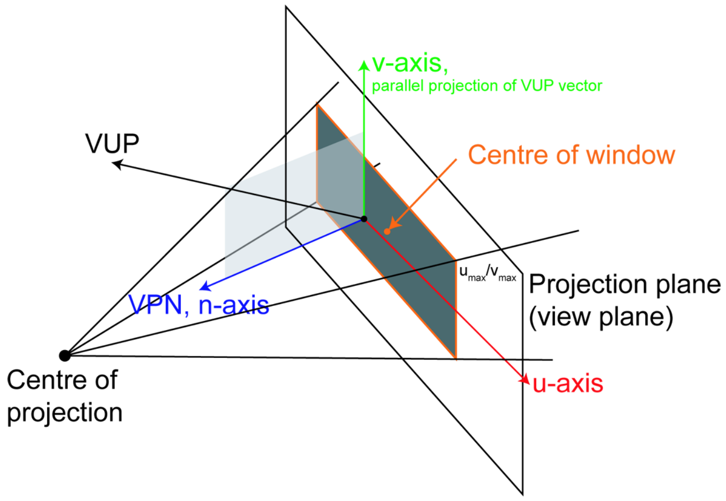


Figure 6.7: View plane with clipping window

Figure 6.8 shows the relation between, the viewing reference coordinate system and the world (Virtual Reality scene) coordinate system. The viewing reference point and, thus the view plane can be anywhere in the 3D Virtual Reality scene.

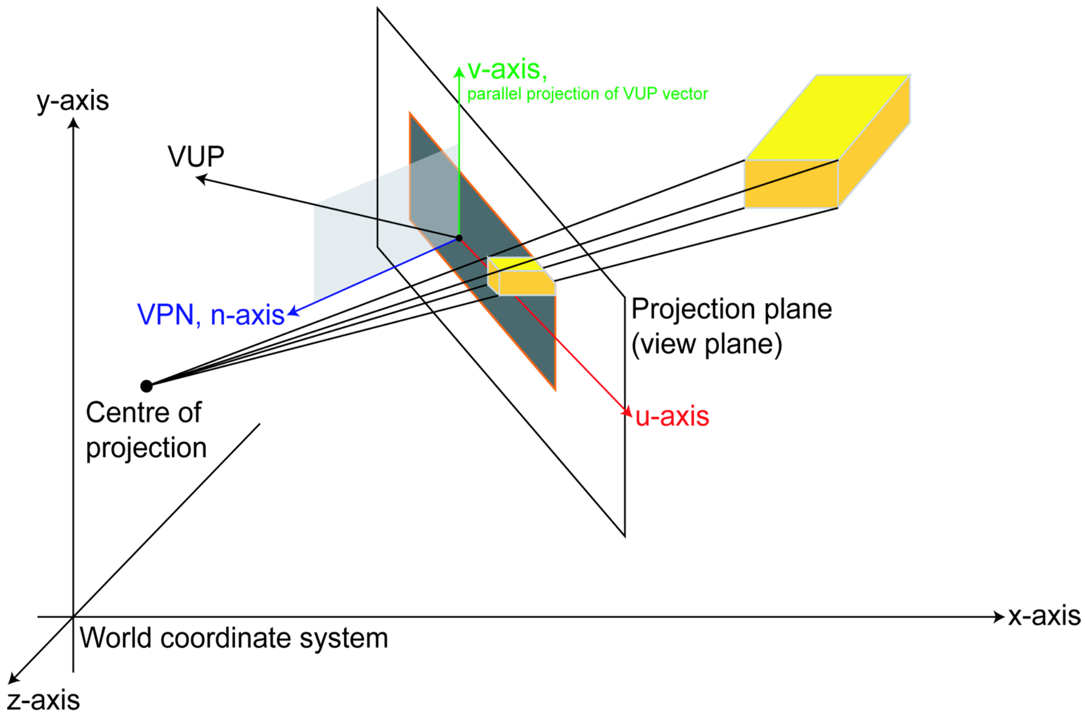


Figure 6.8: View plane, clipping window in real world coordinate system

The centre of projection is not specified in the world. The underlying coordinate system for the centre of projection is the VRC. Thus, the VRP can be moved without any change of the centre of projection.
The clipping with the window discussed above is not the only possibility to limit the scene. In order to clip objects which are relatively near or far from the centre of projection one can introduce a front clipping and back clipping plane. The result is a view volume as shown in Figure 6.9.

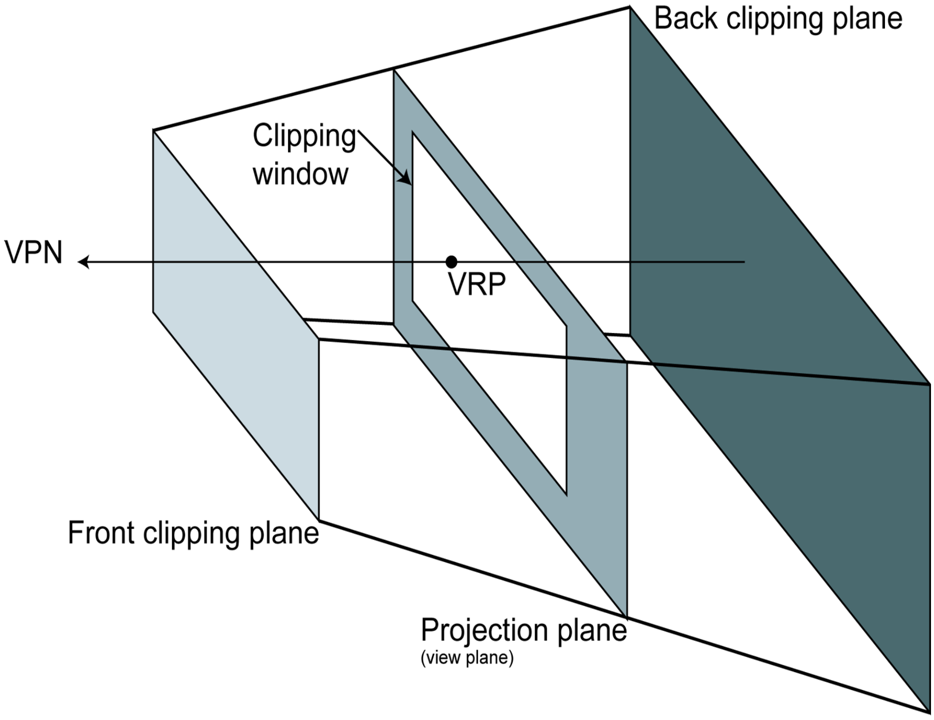


Figure 6.9: Front clipping plane, back clipping plane and clipping window

The reason for the introduction of the back plane may be more obvious than that of the front clipping plane. Objects which are relatively far away are not properly shaped and therefore the object is displayed in a no distinguishable form but the precise calculations have to be done which needs resources. In this case it is better to skip the objects than to display them. Then the user is able to concentrate on the centre of the Virtual Reality scene.
The reason to skip objects near the centre of projection is that such objects tend to look like pick-up sticks, it is not possible to see the shape.
An important point for the generation of the Virtual Reality scene is the detection of the order of the 3D objects. Objects which are in a 3D Virtual Reality behind each other (z-dimension) will overlap in a projected view. Thus, it is important to know which object is in front and which one is in the background. The examples above consist of one object without a complex surface. With more complex examples, the problem of hidden lines and/or surfaces will appear (Figure 6.10).

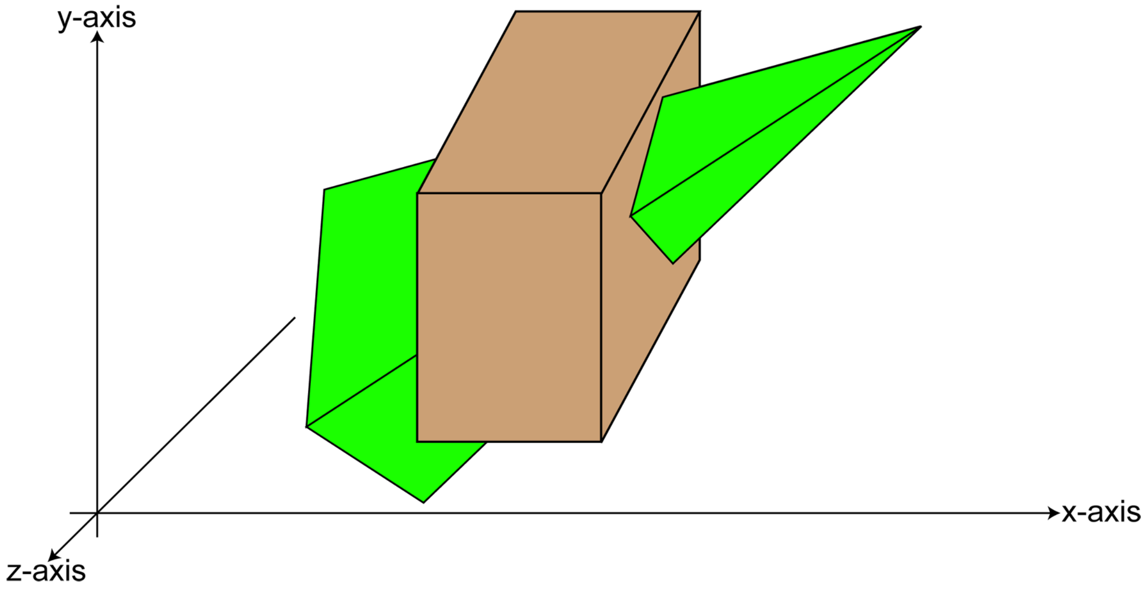


Figure 6.10: Example of hidden line and surface

There are four main algorithms to determine the visible surface of the scene objects [RAU93]:

- z-Buffer algorithm
  This algorithm is developed by E. Catmull
- List-Priority algorithm
  Works fine with less or medium complex scenes.
- Scan-Line algorithm
  Works fine with all scenes.
- Area-Subdivision algorithm
  This algorithm is in general slower than the others.

Because Java3D, M3G and most VRML viewers (e.g. Cortona) uses the z-Buffer algorithm it will be explained in more detail:

An advantage is that this algorithm needs almost the same effort to generate scenes, independent how complex the scene is. Another is that this algorithm works fine with complex scenes. A disadvantage is that it is difficult (high effort needed) to use this algorithm in conjunction with transparent surfaces.

The z-Buffer algorithm needs a buffer with the size of the number of pixels on the screen (resolution 1024x768 needs a buffer size of 786 432). This fact results in a disadvantage of the algorithm, because this amount of values has to be stored. If a floating point value with 32 bit for every z-Buffer value is used, this algorithm needs 3 MB of memory (because of this reason there are some hardware z-Buffer implementations, where the system provides a z-Buffer memory) [FDFH90, FDFH94, RAU93].

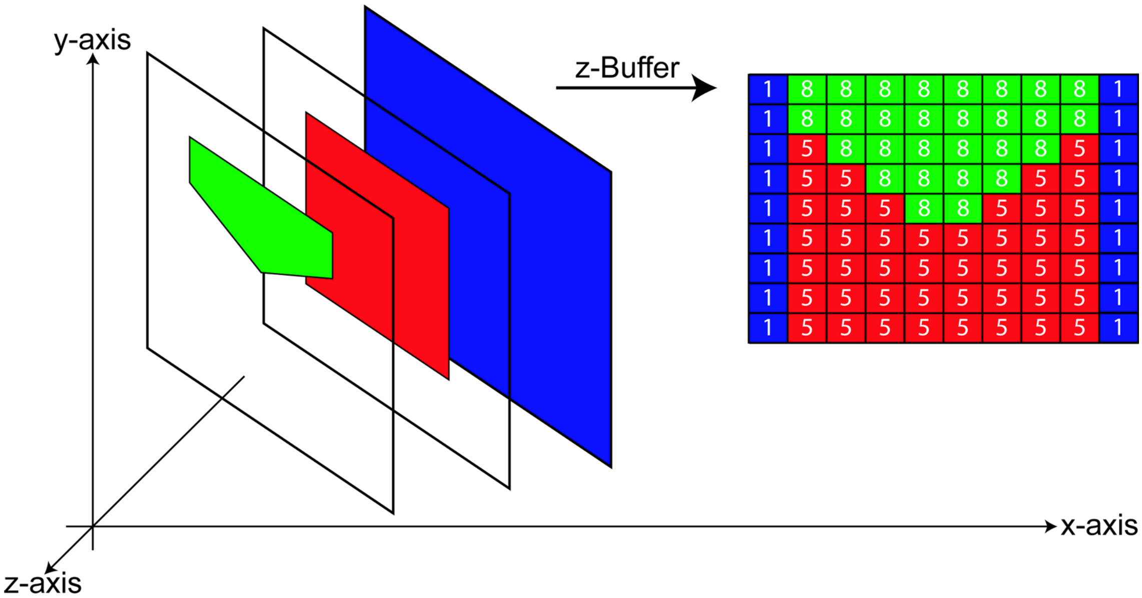


Figure 6.11: Example of a Virtual Reality scene and the corresponding z-Buffer

To obtain the z-Buffer shown in Figure 6.11 the following steps have to be taken: First the whole z-Buffer is initialised with the colour of the background (grey). The z-value for any pixel is anti proportional to the distance from the view plane, i.e. the closer the object is to the view plane, the larger is the z-value. In the next step, one of the objects is taken. If the z-value for the calculated pixel is larger than the z-value in the z-Buffer, the z-value of the pixel is overwritten by the new z-value. This step is repeated until there are no more objects (polygons) in the scene. As mentioned above, this algorithm is not able to handle transparency. At this point it is obvious why. This algorithm only takes into account the colour of one point for the pixel. To work with transparencies, the points behind this have to be included in the calculation for the colour. The algorithm can be improved (regarding the time) if the objects (polygons) are ordered according to the z-value. The algorithm starts with the object (polygon) with the lowest z-value.

6.1.3 High Level Objects (Generation of Mesh)

In order to realise a higher level object such as a cube or sphere either the mathematical equation (vector based) or an approximation of the object can be used. Computer graphics uses in most of the cases for the presentation of such objects a approximation. In 3D graphics the simplest geometrical object is a triangle. Therefore every more complex object is approximated out of triangles. Such a array of triangles is called mesh. Figure 6.12 shows a higher level object (cube) and a mesh representation of this object.

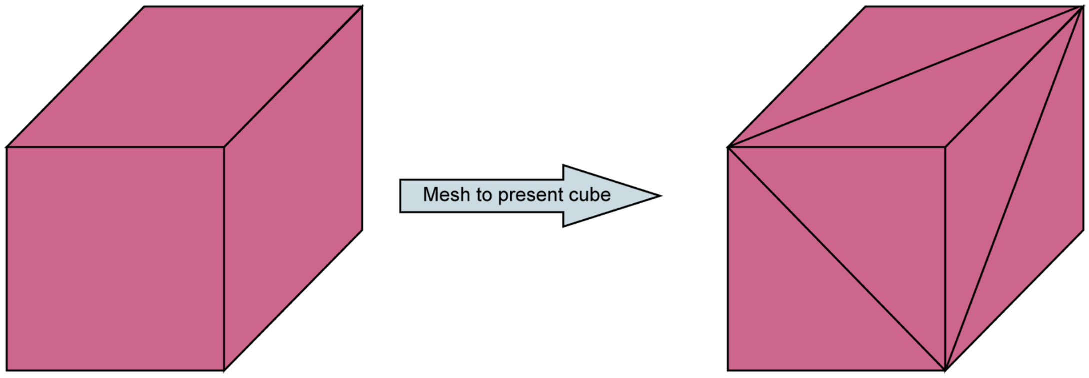


Figure 6.12: Example of a cube mesh

In this case replacing the object by a mesh representation results in the same visual presentation. In this case the approximation of the object is identical to the object by the use of a finite number of triangles. This is different in case a sphere should be approximated by a mesh. Theoretically the amount of triangles has to be infinite in order to produce the same visual presentation (Triangle converge to a point). Practically it is enough if the size of the triangle is smaller or equal to a screen pixel. But this could still result in an enormous amount of triangles. Figure 6.13 shows an example of a reconstruction of a sphere with different meshes.

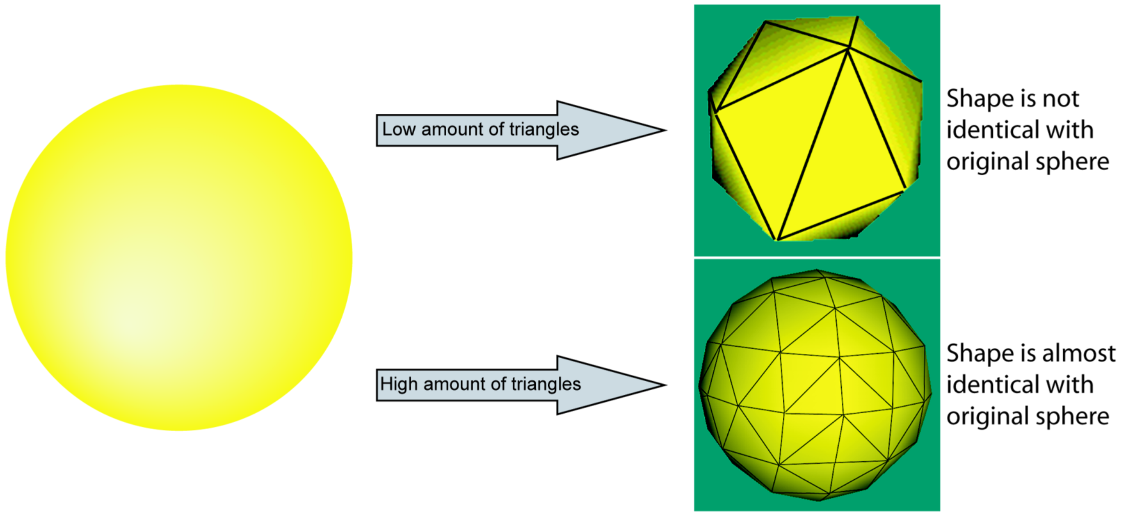


Figure 6.13: Example of sphere mesh

The more triangles are used in the mesh the more the mesh converge to the shape of the original object. As a consequence of the increasing amount of triangles in a mesh the needed processing power to calculate this 3D scene increases as well.

6.1.4 3D Data in Java (Java3D, M3G) and VRML (X3D)

The Java3D (JSR-189) and M3G (JSR-184) specification are managed by the Java Community Process (JCP). Currently the JCP organisation has over 700 members. The members discuss, maintain and pass a so called Java Specification Request (JSR) [JCP]. The VRML and the new successor X3D are maintained by the web3D Consortium [W3C].

There is one important difference between Java3D/M3G and VRML (X3D). VRML is a 3D data description language which specifies the objects. By contrast to that, Java is a 3D programming library (API). Therefore VRML has always need of an additional viewer which does the rendering according to the data description. As a result, Java provides more flexibility for customised behaviour while VRML needs no (or less) programming skills and the developer can concentrate on the design of the virtual world. Java3D does not have an own description language which describes graphical objects as a file format. Thus, so called loaders are available which are able to read a specific file format and to transform it into the internal Java3D object representation. In contrast to Java3D the M3G specifications include a M3G file format description.

In Java3D, M3G and VRML 1.0 both a parallel and a perspective view of the scene can be realised. Because of the similarity with the human visual system the perspective projection is used.

VRML [MC97, RCRR97, VRML]**:**

Figure 6.14 shows the general structure of a VRML scene graph. It consists of non, one or more viewpoints and non, one or more light sources. Geometrical structures (shapes) can be grouped and positioned by group nodes and transform nodes, respectively.

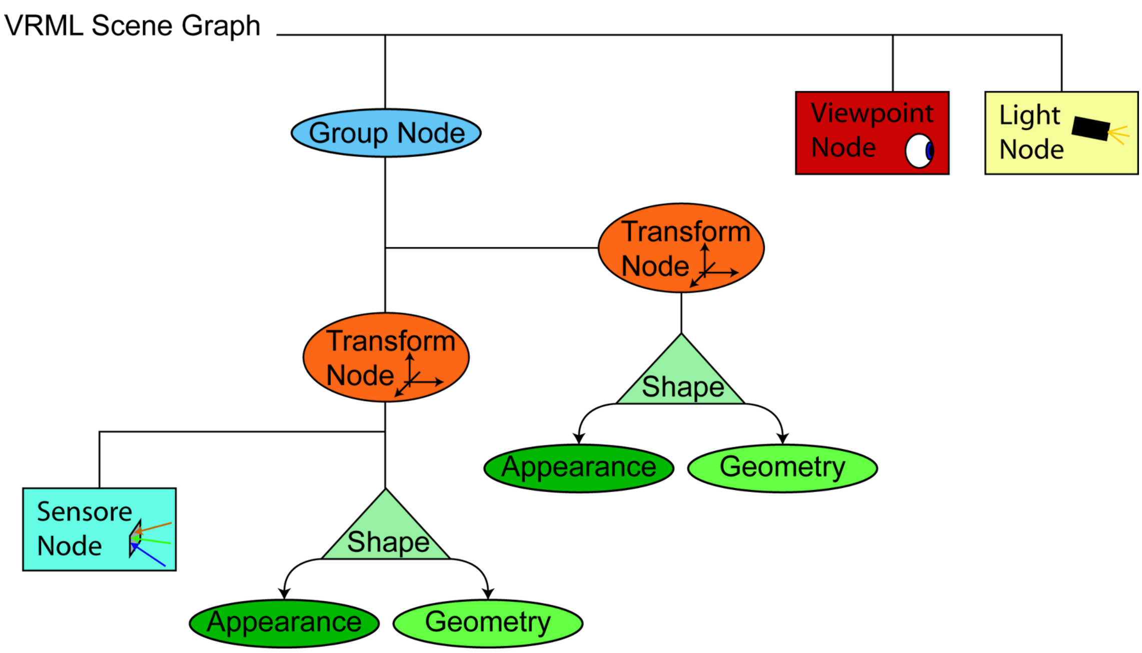


Figure 6.14: Hierarchical structure of a VRML scene graph

Some grouping nodes (e.g. Group, Transform) can be a child of another grouping node in order to realise a hierarchical structure.
Animations are handled by event chains. In such a chain a sender object transfers a value to the corresponding receiver object(s) which update the parameter with the new value.

According to the VRML specification there are three different type of nodes:

- Grouping Nodes: Group, Switch, Transform, etc.
- Children Nodes: Shape, Light, Transform, etc.
- Nodes which are not valid as children nodes (complex field values): Box, Appearance, Material, etc.

Java3D (JSR-189) [BOU, J3ASE]**:**

The general Java3D scene graph shown in Figure 6.15 consist of a main BranchGroup node. All other objects are added to and non, one or more light sources and geometrical structures (shapes). The grouping of different nodes can be realised with TransfromGroup nodes and BranchGroup nodes, respectively.

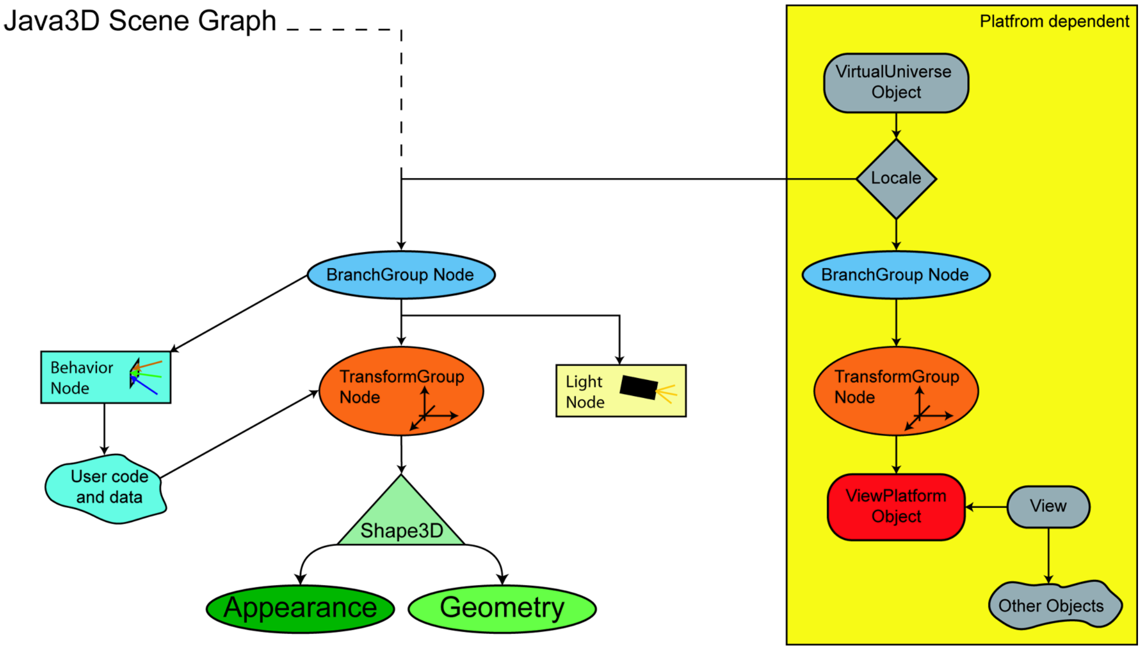


Figure 6.15: General Java3D scene graph hierarchy

Animations are performed with *Behavior nodes* in which the user can add customised code to adapt the behaviour for his needs. Thus, any accessible value of an accessible node can be manipulated.
Java3D stores a Virtual Reality scene in a so-called *Locale Object* which is a high resolution coordinate system. The *Locale Object* is included in a *VirtualUniverse Object* which can contain several *Locale Objects*. In this way Java3D can handle several Virtual Reality scenes in one application. The yellow highlighted part is a platform dependent implementation. The *ViewPlatform* can be influenced by

platform dependent features as well as by other additional (possible platform dependent) objects such as input devices, tracking devices, etc. It is possible to introduce new objects by extending existing ones.

M3G (JSR-184) [M3G, NDF, SEDF]**:**

The M3G specification has a low amount of available objects. There are no higher level object available (e.g. box, cube sphere, etc.) The implementation of new higher level geometrical objects have to be realised by extending a *Mesh* object. The general scene graph of M3G shown in Figure 6.16 consist of a World node, a Background node, Group node, Transform nodes, one camera, one or more light sources and geometrical objects (Mesh nodes).

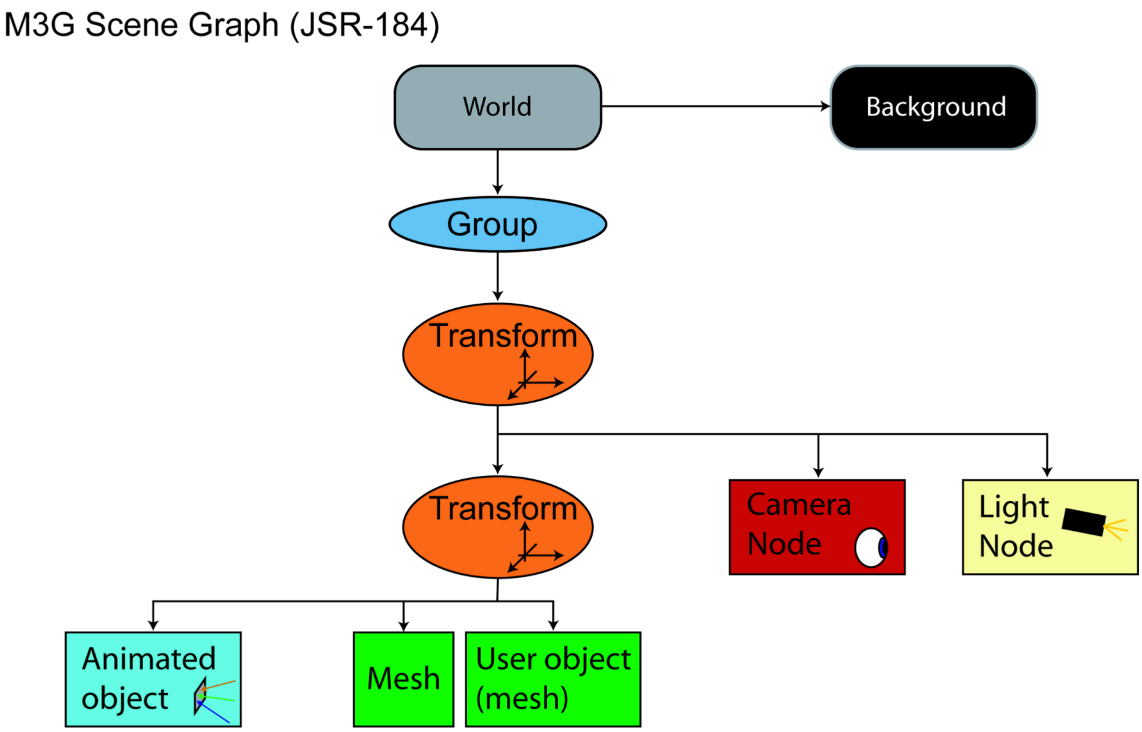


Figure 6.16: Scene graph hierarchy of M3G

The M3G concept of animation does not allow customised code in an animation. Therefore only values and nodes marked as animatable can be manipulate. This fact limits the possible animation. However, the most important values and nodes are covered, thus programmers are not really limited most.

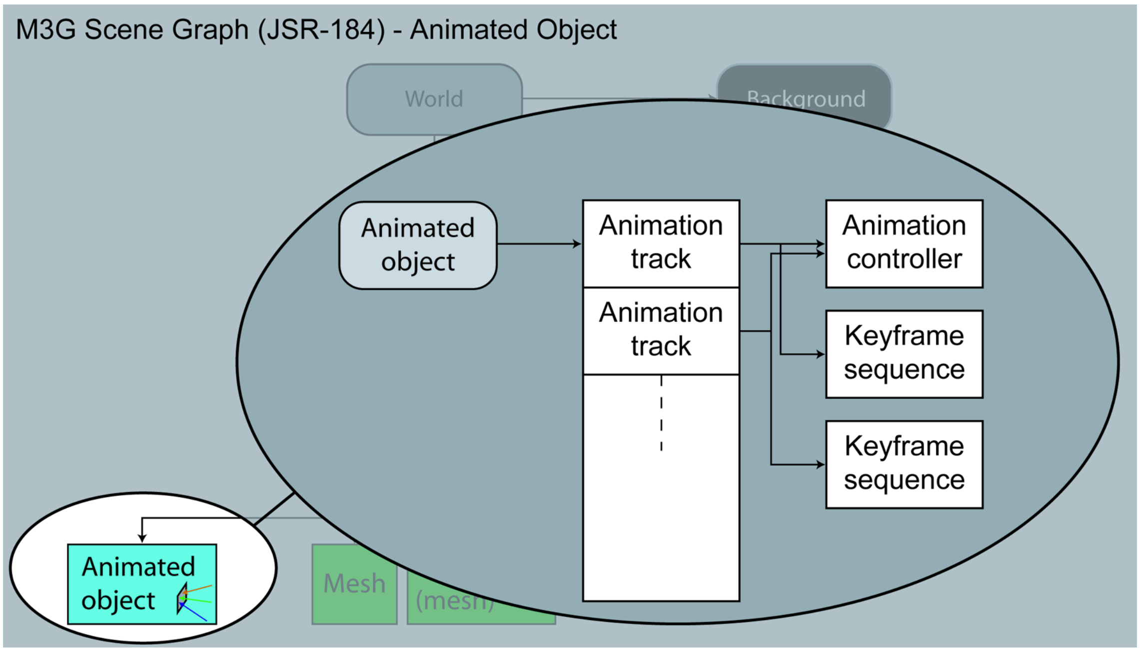


Figure 6.17: Animated in M3G

All three specifications use a similar scene graph concept and vocabulary. For example the *Shape* node (Figure 6.14) in VRML is called *Shape3D* (Figure 6.15) in Java3D. In order to position grouped objects in the virtual three-dimensional space a transforming group object with a Transform Node (TransformGroup) is used. To realise a hierarchical structure all nodes be a child of only one group. However, loops of grouping is in all three concepts prohibited (Figure 6.18).

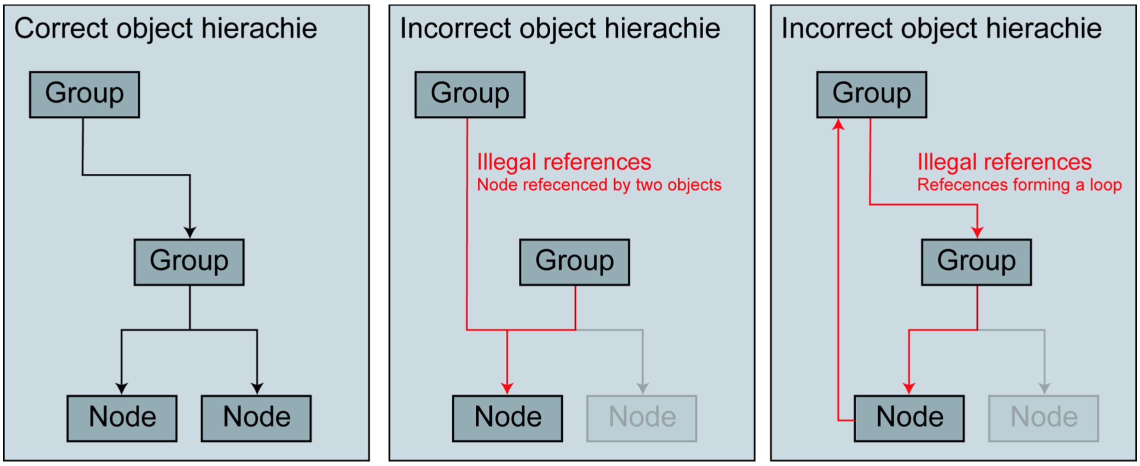


Figure 6.18: Hierarchical structure, multiple parents and looping is prohibited

The VRML and Java3D specifications use the segmentation of the Shape (Shape3D) Node into *Geometry* and *Appearance*. This is not transferable to the M3G specifications because in this specification high level shapes are not available. In the M3G case the programmer has to generate high level geometry objects such as boxes, spheres, etc. with the help of a *Mesh* node which reuses the concept of the Appearance node.

Java3D and M3G are more powerful concerning user defined features because these specifications are embedded into a programming language. Hence, the possibilities of manipulations of a Virtual Reality scene are almost unlimited.

VRML offers less possibilities for user defined manipulations but thanks to the more generalised structure programmers can concentrate on the realisation of the Virtual Reality content.

6.2 3D Data Visualisation as Monoscopic View

A “Monoscopic View” is defined as the presentation of a three-dimensional Virtual Reality on a two-dimensional screen. The standard mobile device screen is like a standard CRT or TFT monitor: It is unable to display a Virtual Reality scene in three-dimensions because the monitors are only two-dimensional. For this purpose the three-dimensional Virtual Reality scene has to be reduced by one dimension. In this project the work flow to perform this task is the following, given a file with a 3D Virtual Reality scene:

1. File is transferred to mobile device
2. File is parsed into internal M3G representation
3. M3G library renders image on base of the specified projection method
4. Resulting two-dimensional image is displayed on screen

In order to generate a monoscopic view the most important steps in this work flow are first to establish an internal software object representation of every graphical object and second to render this internal software objects into an two dimensional image. The first step is performed with the help of a parser module initial written in the framework of a master thesis and improved by a project work. The second step is realised with the help of the M3G library.

To set-up the rendering engine correctly following parameters have to be specified:

- Camera position
- View volume (Front- and back-clipping)
- Camera aspect ratio
- Viewport dimensions (clipping window)

The camera is a scene graph node that defines the position of the viewer in the scene and the projection from three-dimensional to two-dimensional. The camera is always facing towards the negative z-axis in its local coordinate system. The camera can be positioned and oriented in the same way as any other node. The projection matrix transforms homogeneous coordinates from camera space to clip space. Triangles are then clipped to the view volume, which is defined by the width, height and depth of the view frustum denoted by w, h and d, respectively. The perspective projection matrix is constructed out of this values.

After clipping, x, y, and z are divided by the values of the projection matrix to obtain normalized device coordinates (NDC). These coordinates are between minus one and one and the centre of the viewport lies in the origin. Finally, the viewport mapping and the depth range are applied to transform the normalized x, y and z coordinates into window coordinates.

The image obtained by the rendering engine can either be displayed on the screen or drawn to an off-screen image. In case of screen rendering, the screen is blocked during the writing of data to the screen. In case of off-screen rendering the rendering engine uses an image to write the rendering data. Using the off-screen rendering method the problem of flickering screens can be solved. The flickering is caused by an underlying system process which starts to read the incomplete image of the rendering engine.

6.3 3D Data Visualisation as Stereoscopic View

6.3.1 Generation of Stereoscopic View

The “Stereoscopic View” is defined as a presentation of a 3D Virtual Reality on a two-dimensional screen. This technique is capable of recording three-dimensional visual information or creating the illusion of depth in the image. Thus, the viewer gets a more realistic impression [HOL05, HS3W, SJK06].

The depth information is produced by the human visual system and the brain. The human visual system gets slightly different images for the left and the right eyes respectively. The brain combines both images and in this way obtains the depth information (Figure 6.19).

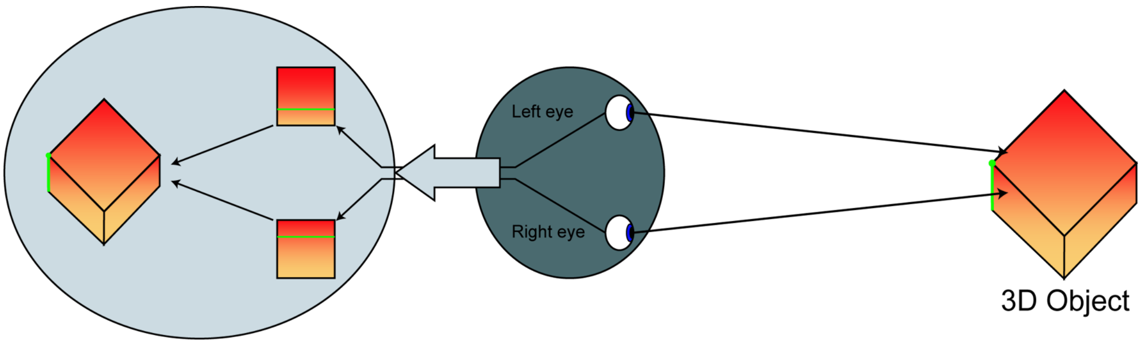


Figure 6.19: How humans see 3D objects, realisation of a stereoscopic presentation

In order to transfer depth information from a two-dimensional screen or image to a human viewer the system has to provide two slightly different images for the left and the right eyes. Therefore, the process of generating a stereoscopic view can be derivative from the process of generating a monoscopic view. To do so such a system additionally needs the information of the virtual “eye distance”. Knowing the eye position the camera position for the left and right eyes can be calculated. Such a system performs a rendering with the left camera position and additionally with the right camera position. The result is an image for the left and the right eyes respectively.

The virtual “eye distance” is an important value. It partly defines the quality of the resulting stereoscopic view. The centre of the eyes, one eye and the object itself form a triangle. If the aspect ratio of this triangle in a real scene is significantly different from the aspect ratio in a virtual scene the resulting image will be confusing for the human viewer. Since the projected object and the depth information does not match up. The human brain unsuccessfully tries to match up this information.

After such a system has generated the two different images for the eyes it has to ensure that the left eye sees the left image and the right eye sees the right image, respectively. For this purpose the following methods are available:

- Build in screens (with aperture system, parallax barrier)
- Lenticular lenses
- Anaglyph system (colour filtered images)
- Shutter systems (shutter glasses)
- Polarised light systems
- Head Mounted Displays (HMD)
- Holography

All technologies mentioned above, can be used for non static three-dimensional presentation. An exception is the holography. The traditional holography technology allows static images to be viewed in a three-dimensional way but it is not possible to record moving images like an animation, film or interactive three-dimensional scenes in contrast to the other technologies. The next chapters explain these techniques in more detailed. Further it will be discussed which technology can be used on or in connection with a mobile device.

6.3.2 Output Device: Mobile Device Screen

In order to generate a stereoscopic view on a mobile device screen, the following three possibilities are usable:

- Build in three-dimensional screen
- Lenticular lenses
- Anaglyph images

Build in three-dimensional screen [H3DW, LCD04]**:**

During the last few years more and more so-called 3D monitors are available on the market. All of them has the common principle of providing different images for the left an the right eyes, respectively. This is mainly achieved by aperture system (parallax barrier).
In mid/end 2004 Sharp brought the first mobile phone (SH 505i) designed for NTT DoCoMo with three-dimensional screen on the market. Up to now there are no mobile phones with three-dimensional screens available for the European market.
Figure 6.20 shows the mode of operation of Sharp's three-dimensional LCD screen. Between the back light and the TFT-LCD there is a “switching liquid crystal” layer - the parallax barrier - which can be turned on (blocks light) or off (light can pass through). In the case that the crystal layer is switched on it blocks the transmitted light from the back light in such a way that the right eye receives light from TFT pixels designed for the right eye and the left eye receives light from TFT pixels designed for the left eye, respectively.

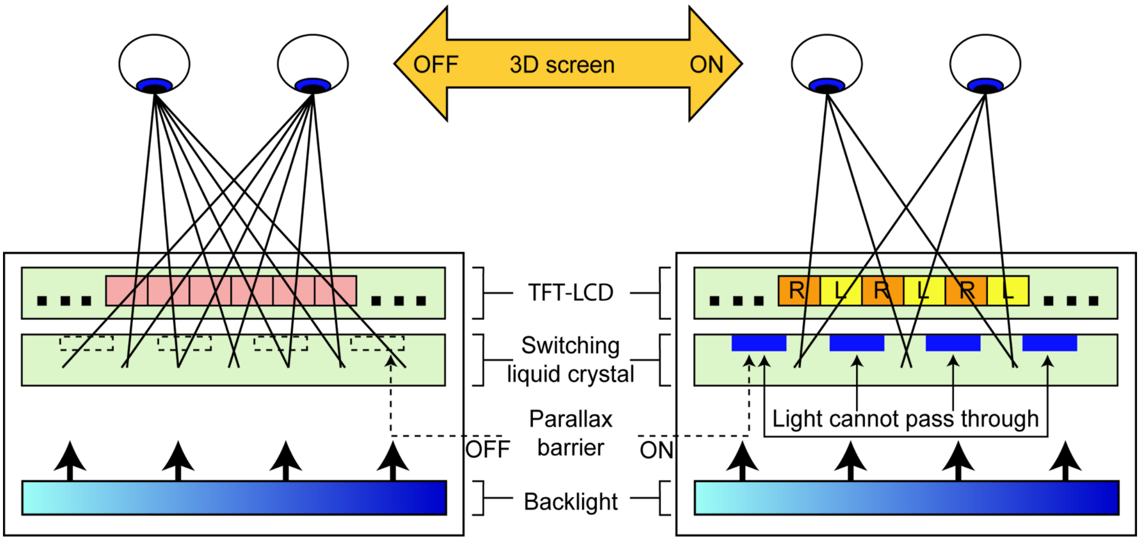


Figure 6.20: 3D screen with parallax barrier method

This principle is limited to a specific view range and angle. Figure 6.21 shows an example of an eye position which is too close. The result is that one eye can see some pixels designed for the other eye (red light ray). The same is true if the eye position is too far or the angle is too large (eye position too left or too right) [LCD04].

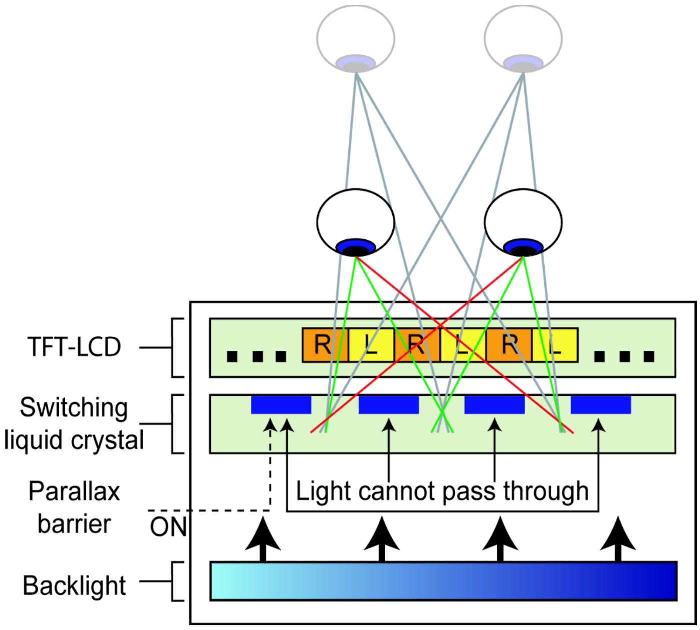


Figure 6.21: Eye position out of range for stereoscopic presentation

Advantages of this system are:

- Three-dimensional view can be turned on/off
- Available software library
- No need of additional glasses or other devices

Disadvantages of this system are:

- Limited view range (distance and angle)
- Not available in Europe (only available from Sharp)

Lenticular lenses [WHE03]**:**

Lenticular lenses are sheet of parallel cylindrical lenses. The mode of operation is similar to the build in three-dimensional screen. It uses the principle of parallax barrier. The lenticular image has to be designed in such a way that each eye's line of sight is focused on a different strip (see Figure 6.22).

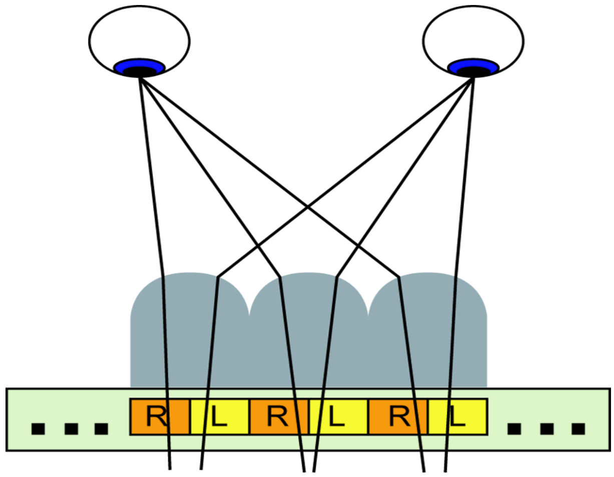


Figure 6.22: Stereoscopic presentation with a lenticular lens

The original image of the right and left eyes are sliced into small stripes. Then these stripes are interleaved into one image. Figure 6.23 shows an example of such an image.

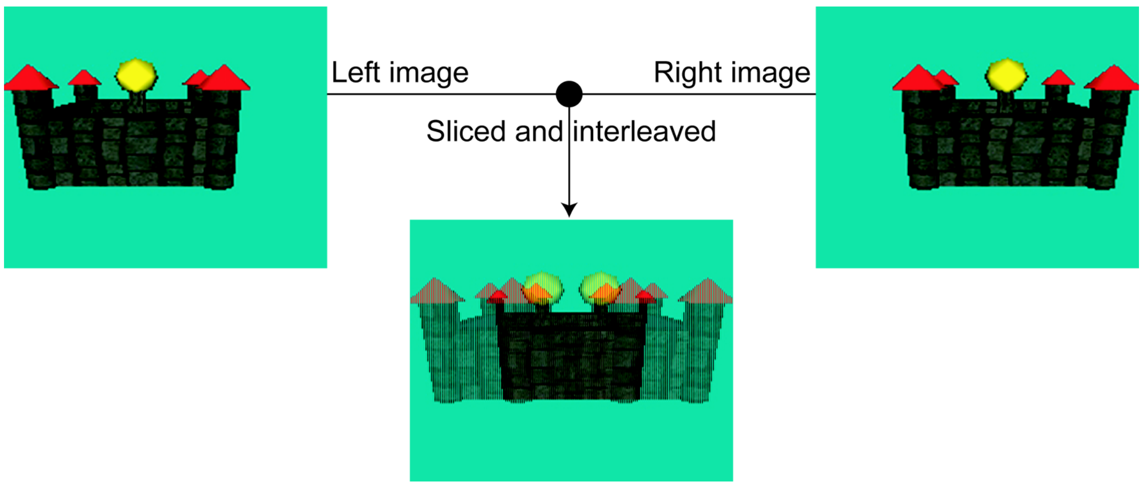


Figure 6.23: Slicing and interleaving right and left images into one image

The width of the stripes and the radius of the lenses have to fit together in order to create a three-

dimensional impression. In the case that such an image is shown without lenticular lenses, the viewer will get a strange, blurred impression.
This technique has similar limitations for the view distance and angle because it uses the same principle (parallax barrier). If the technique is used in connection with a monitor (TFT or CRT) an additional pre requirement has to be taken into account. The order of the RGB-colour pixels has to be in a line and must have the same orientation (portrait or landscape) as the lenticular lenses. If this is not the case colour distortion will occur.

Advantages of this system are:

- No need of additional glasses or other devices

Disadvantages of this system are:

- Limited view range (distance and angle)
- Can not be turned on/off, i.e. non three-dimensional content will look strange
- No software library available
- Mobile device screen has to fulfil specific requirements

Anaglyph images [PASI]**:**

An anaglyph image is the colour filtered combination of the image for the left eye and the right eye. To obtain such an image, one has to filter the image for the right eye, for example, with a red colour filter and the image for the left eye with a cyan colour filter. Commonly used filter combinations are listed in Table 6.1.

Right eye	***Left eye***
cyan	red (best colour reconstruction)
blue	red
green	red

Table 6.1: Commonly used anaglyph filters

Both images are merged together into one image. In order to obtain the three-dimensional depth, the viewer has to wear glasses with the same filters. In this way the right eye views the image originally obtained for this eye and the left eye views the corresponding original left image.

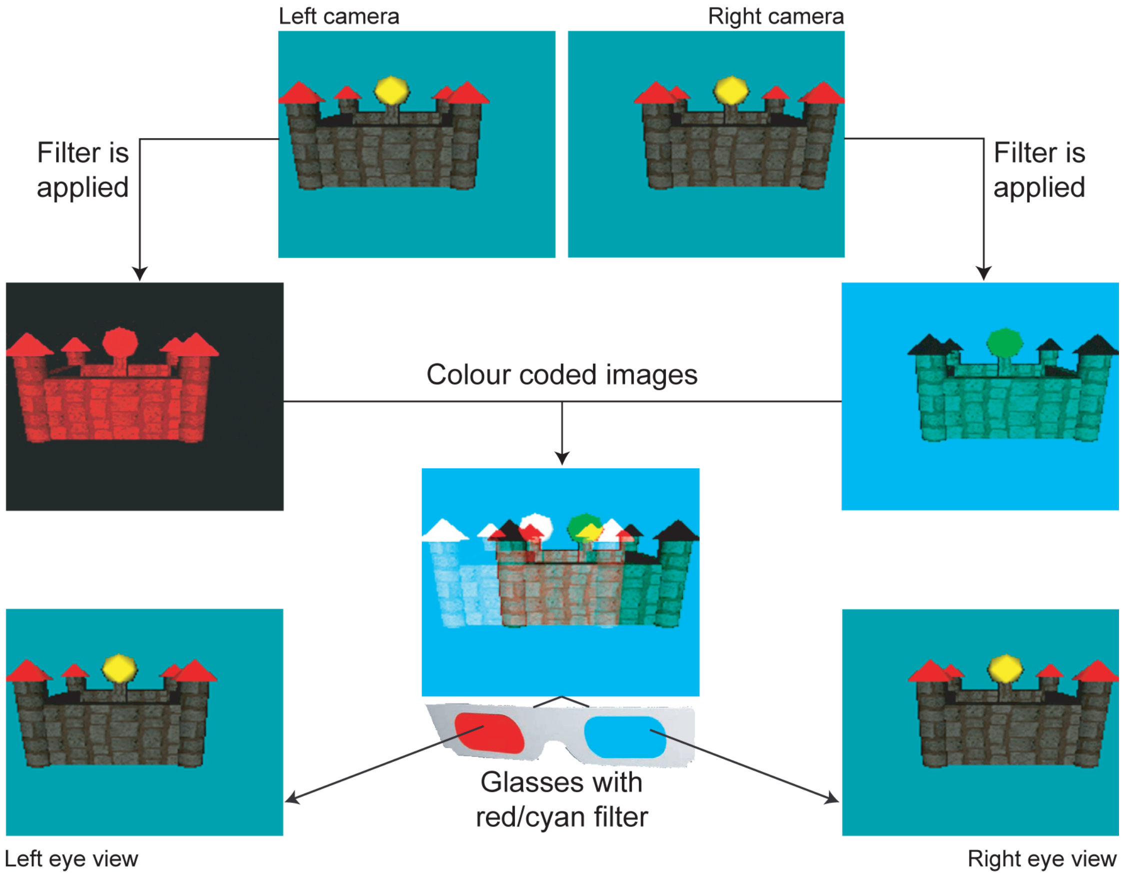


Figure 6.24: Analglyph image generation

Problems occur if the original images contains colours close to the filter colours. By using the combination of red/cyan filters it is possible to obtain reasonable (but not accurate) blue sky, green vegetation and appropriate skin tones. Colour information appears disruptive when used for brightly coloured and high contrast objects which contain colours close to red or cyan. The worst case occurs if an image contains big objects in red or cyan. In this case, the method will give bad results. Since the human eye is less sensitive to the red colour than to the other primary colours, the viewing of an image is improved if the non-red colours are made less intensive in brightness. Viewing the images through glasses will increase the dimness because the filters are reducing light intensity. To compensate this effect, the overall brightness and contrast can be increased.

Advantages of this system are:

- Method is applicable for all kind of screens
- Anachrome "compatible" method allows viewing without glasses

Disadvantages of this system are:

- Additional equipment is needed (glasses)
- No software library available
- On small screens image tends to be blurred

6.3.3 Output Device: External/Additional Devices

The remaining technologies (Shutter system, polarised light systems, HMD, holography) to generate stereoscopic views, are not or only with too much effort applicable on mobile devices. Nevertheless, these methods can be used in connection with external or additional devices and systems, respectively. These external and additional devices and systems are limiting the user's mobility depending on the device or system complexity.
Another technology is the method of shutter systems. This method has not been tested in this research. However it is an important technology hence it will be shorty discussed.

Shutter system [SCE]**:**

This system is used in connection with so called shutter glasses which are synchronised with the display system either with a cable connection or wireless. Standard systems are working with an overall refresh rate of 60 Hz whereas the image is switched every time between the left and right eye view. The shutter glasses are synchronised in such a way that the glasses for the right and left eye are switched between transparent and non-transparent corresponding to the displayed image. Unfortunately, most of the TFT and LCD displays cannot be used because of the luminescence. This reason in addition to the difficulty to receive refresh rate information from mobile device screens makes it impossible to use this technology directly on mobile devices. However this method can be used in connection with a personal computer based display system which uses the shutter technology. The mode of operation will be similar to the functionality of the video projector system. This system is discussed in the following paragraph. Figure 6.25 shows an example of commercial shutter glasses.

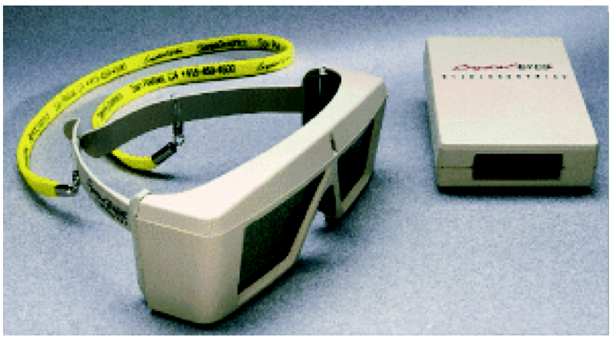


Figure 6.25: Commercial (Stereographics, Crystal Eyes) shutter glasses

Video projector system [LSP]**:**

The video projector system consists of two video projectors which use polarised light to provide two different images to the viewer. The projector for the right eye view is filtered with a horizontal polarised light filter and the left eye view is filtered with a vertical polarised light filter (or visa versa). Both beams are displayed on a special video projection wall which has to reflect the light without to change the polarisation direction. A user wearing glasses which have polarised light filters in the same way as the video projection system will see with the right eye the right image and with the left eye the left image. The practical realisation of such a system will be discussed in more detail in Chapter 6.4.3.

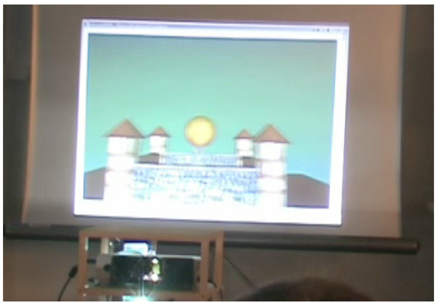


Figure 6.26: Video projection system (LSP)

Head Mounted Display (HMD) [LSP]**:**

A Head Mounted Device (HMD) consists of two screens which are mounted in front of the eyes. There exist two groups of HMD; namely the mono (two-dimensional) and the stereo (three-dimensional) displays. The two-dimensional HMD uses the same input signal for the right and the left screen in contrast to the three-dimensional HMD which has separate input signals for the right and the left screen, respectively. Therefore, the second group can be used for stereoscopic presentation. This group uses different methods to receive the two different signals for the two different eye views. The following signal input methods are used:

- Dual input (two input connector for VGA, S-Video)
- Interlace
- Page flipping
- Sync doubling (split screen)

Figure 6.27 shows an example of a commercial HMD and Figure 6.28 shows the prototype developed by the Laboratoire des Systèmes Photoniques (LSP).

Figure 6.27: Commercial HMD

Figure 6.28: HMD prototype developed by the LSP

6.4 Integration into the MediaPhotonics Project

6.4.1 Introduction to MediaPhotonics

Figure 6.29 shows an overview of the MediaPhotonics project which is developed by the Laboratoire des Systèmes Photoniques (LSP) and the University of Applied Sciences Offenburg (FHO), as a subproject in the Rhenapotonics project.

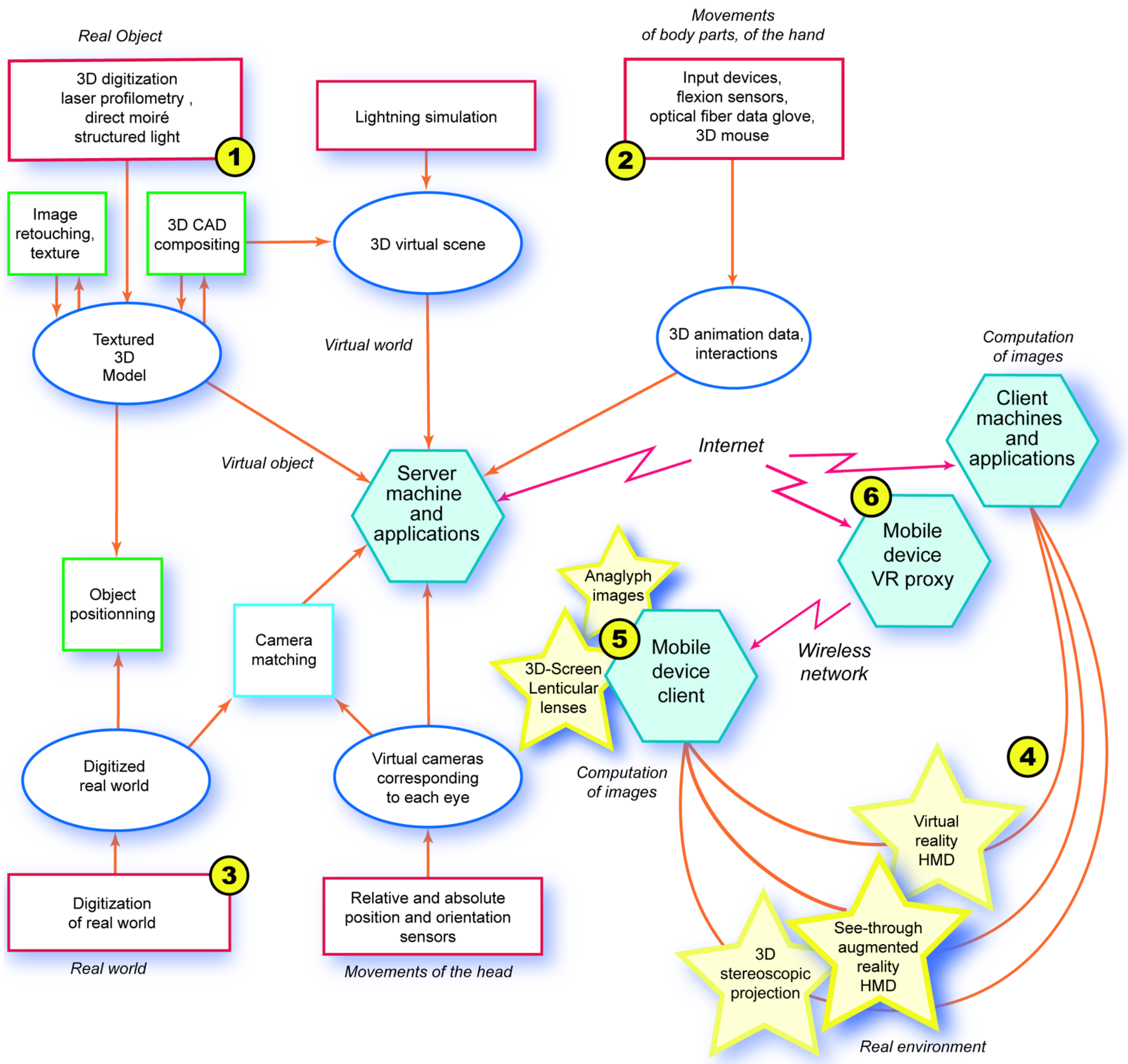


Figure 6.29: Overview of the components of MediaPhotonics project

1. Project LSP [LSP]
2. Project LSP [LSP]
3. Project Thomas Wendling (see Chapter 6.4.4)
4. Video Projection System LSP (see Chapert 6.4.3)
5. Visualisation of VRML/X3D on mobile devices [JER04, KOS04, ALLS05]
6. Optimisation of VRML/X3D data for mobile devices

6.4.2 Integration this research into MediaPhotonics

The MediaPhotonics project is a study on concepts of an augmented mobile visualisation system. Studies on how real and virtual objects can be brought together in order to provide more information to the end user. Studies are done for:

- New solutions for input and output
- Positioning
- Object recognition and synthesis of virtual worlds
- Mobility

This research project is focused on the mobility aspect. It is integrated in the MediaPhotonics project in such a way that it is acting as middle ware to connect the server machines and applications with the output systems. Figure 6.30 shows the point of intersection between this research and the higher-level MediaPhotonics project.

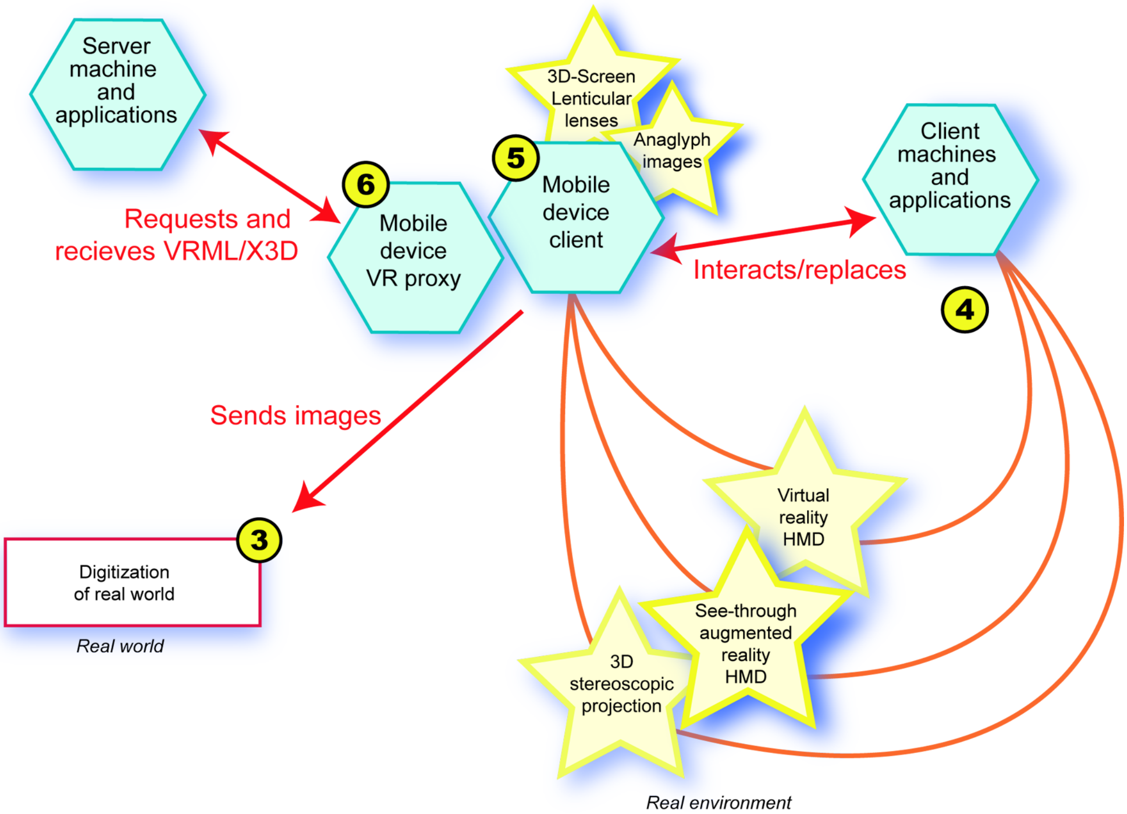


Figure 6.30: Integration of the work into the MediaPhotonics project infrastructure

Since the connecting device is a mobile device the users are gaining mobility which is more or less limited by the output system. To overcome this problem, the mobile device has to be used as well as output system. This way it will partly replace the personal computer based client machines and applications. In order to provide a feedback, the mobile client device can send images (or a video stream) to the MediaPhotonics system which will generate a Virtual Reality out of those data. This component is important for object recognition and to be able to place real objects in virtual world or virtual object into the real world. All these components together are a step towards an augmented mobile visualisation system.

6.4.3 Displaying 3D Data on a Video Projection System

A video projection system generates a stereoscopic presentation using of polarised light filters. Users equipped with glasses with polarised light filters are able to see the two different images with the corresponding eyes. Thus, their brain can generated the depth information in order to get an impressive three-dimensional view. Figure 6.31 shows the principle system components needed to generate such a three-dimensional illusion.

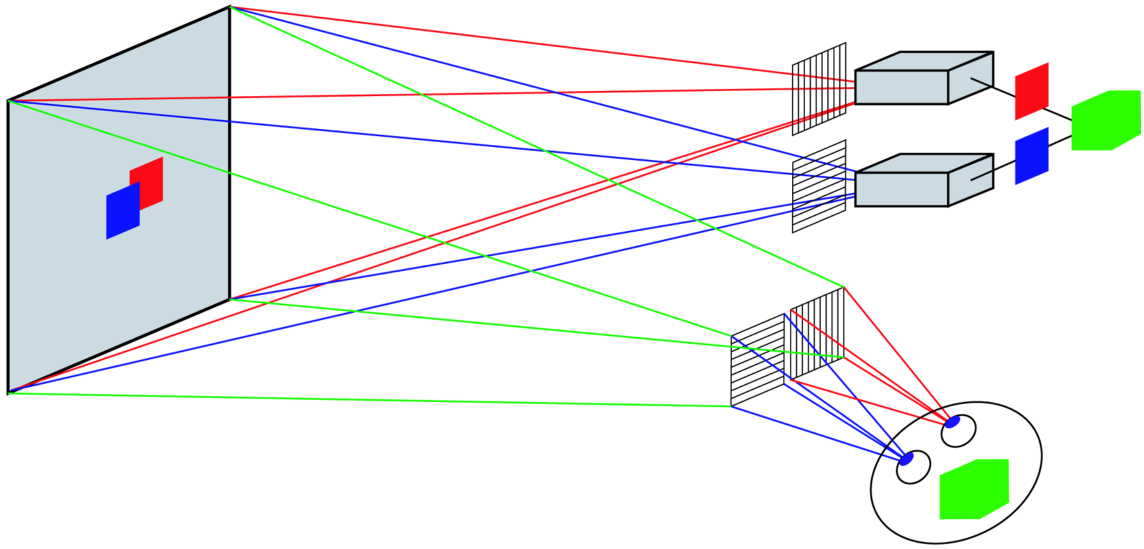


Figure 6.31: Stereoscopic video projection system

The 3D Virtual Reality scene (green cube) is split into two slightly different images (red, blue). The difference of the two images is a result of different camera angles. The images (video streams) are sent to the corresponding video projectors. The light beam of every video projector is filter with a polarised light filter, one with a horizontal polarised and the other one with a vertical polarised light filter. The resulting image on the projection screen is a overlap of both images. Therefore, users without glasses will get irritated by the strange image. Nevertheless users equipped with polarised glasses will see the correct image. In order to be able to work in this way, the projection screen has to reflect the light without changing the polarisation angle. Otherwise this method will not work because the images will interfere. In this case, the polarised glasses will not be able to separate the images correctly. In the case that the projection screen does not change the polarisation angle the polarised glasses are able to separate the two images for the corresponding eyes.

6.4.4 Towards an Augmented Reality Mobile Visualisation System

The system developed by Thomas Wendling is able to reconstruct a 3D Virtual Reality scene with the help of a sequence of photos from the scene. In order to obtain a 3D Virtual Reality scene, the following six steps have to be performed [WEN04]:

1. Calibration of the camera
2. Correction of desorption
3. Detection of camera position
4. Definition of the basis plane (equitation in space)
5. Incorporation of basic volumes
6. Adding texture to the volumes

The calibration of the camera has to be done once as long as the lens remains the same. The developed software is able to recalculate important points of the scene with the help of step number two up to four. Step five generates a 3D Virtual Reality scene with the help of basic volumes (Cube, sphere, cylinder, etc.) the shape of the 3D Virtual Reality scene is in the ideal case the same as the original scene. In order to generate a more realistic 3D Virtual Reality scene the objects are mapped with textures obtained from the images.

The result of step number four is very important because with its help it is possible to perform an object recognition. An important issue of an augmented reality mobile visualisation system is the ability to enrich the real world with additional virtual objects (information). In order to add additional virtual objects or to provide information to real objects one has to know the position of the object in the real scene. For example, to be able to place a virtual cup on a real table the knowledge of the position of the table is of essential importance. The developed software provides basic knowledge of important positions in the real scene. Thus it is possible to detect objects and their position.

However it is yet not possible to completely integrate mobile device in this reconstruction process. This is due to the mobile device camera. The resolution of a mobile device camera is not that high as the resolution of a standard digital camera. Since a camera for a mobile device has to be cheap in price the used lenses are not of high quality. This fact adds more desorption to the image than it would be the case with a standard digital camera. Thus, the quality of the images is reduced which makes it difficult to use these images to reconstruct a 3D Virtual Reality scene. Nevertheless, it is an important step towards integration of mobile devices into the whole project and thus to realise an augmented reality mobile visualisation system.

7 3D Data Processing on the Mobile Device Client

7.1 Architecture Overview

The task of the mobile device client is to request the 3D Virtual Reality data file (VRML) from the VRML proxy. Within the request the mobile device specifies the device (brand and model) name and additional user settings which differs from the standard settings. This information is used from the VRML proxy to request the 3D Virtual Reality data file(s) from the Web server and optimise the data for the corresponding mobile device before sending the data out. The actual client implementation supports the following objects and functionality:

Object	***Detectable/moveable***	***Texture***
Cube	yes/yes	supported
Sphere	yes/yes	supported
Cylinder	yes/yes	supported
Cone	yes/yes	supported
Background	yes/yes	-
Camera	no/no predefined	-
Light	yes/no	-

Table 7.1: List of supported VRML (X3D) objects

After the data is received the information has to be parsed and a 3D presentation of the 3D Virtual Reality data objects has to be generated. The visual correctness of the 3D object depends on the mobile device and the user settings respectively. Figure 7.1 shows the position of the VRML client in the overall system and the general system architecture.

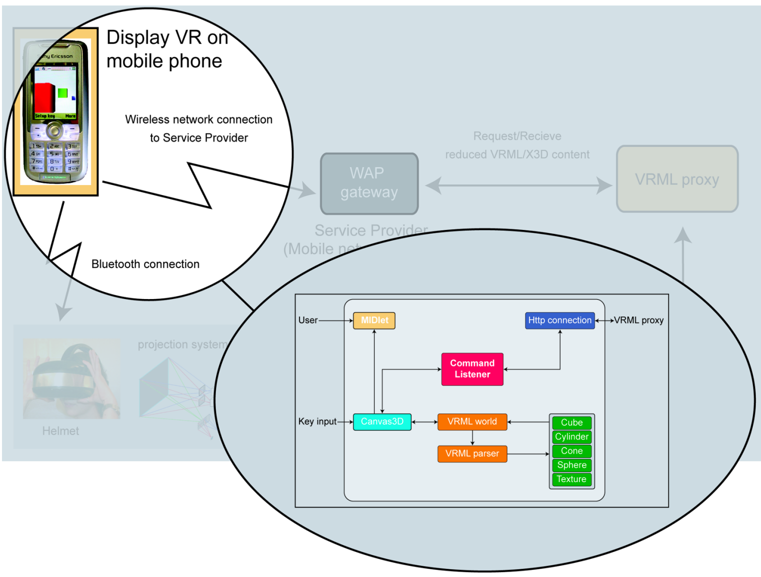


Figure 7.1: System overview with focus on mobile client

7.2 Transmission Data Decompression

The decompression of data is realised with the GZIP method (Chapter 4.2). Since the VRML proxy compresses data with the GZIP algorithm, this algorithm has to be used at the client side as well in order to decompress data compressed and send by the VRML proxy.

The request generated by the mobile device client is not compressed. There are two reasons for this:

- No library available
- No significant advantage if request is compressed

Unfortunately, GZIP is unsupported by the standard J2ME edition, thus the compression method has either be implemented by the author or a third-party library has to be used. Research on the Internet had shown that at this time (May 2006) there were no free libraries for GZIP compression available for the J2ME platform.

Additional studies had shown that there is no or not significant advantage in compressing a request. The reason for this is that a request body contains little amount of data. The only amount of data contained in the request body are the mobile device name and the user settings or a session ID. In this cases compressing the request body does not lead to a reduction of the data amount. In some cases the amount of compressed data is even bigger than the amount of uncompressed data. Therefore, no implementation of a GZIP compression library was made. In order to decompress a response the following procedure is performed:

- Check header if stream is compressed (content type has to be of mime type "application/x-gzip")
- Retrieve the body of the request (as binary data)
- Construct a GZIPInputStream to decompress the data

Because of the missing GZIP support the third-party library from Carlos Araiz [ARA] is used for decompression in a slightly modified version.

7.3 Reduction of Visualisation Data

In order to decrease the time needed to load a 3D Virtual Reality data several optimisation methods were discussed. Beside the reduction of visualisation data through removing objects from the scene, there is another possibility which is directly addressed to the mobile device client.

In Chapter 6.1.4 it was discussed that there are no high level objects available in M3G. Every higher level object has to be generated by the use of a mesh. The more triangles in a mesh are used the more accurate the object is approximated. As a consequence of the increasing amount of triangles in a mesh the needed processing power to calculate this 3D scene increases as well. This disadvantage of the increasing processing power has shown dramatically for the initial calculation of the 3D scene. The reason for this is that the duration of the initial calculation is a few seconds (dependent on the amount of objects). The duration of the calculations to manipulate the scene (zoom, rotate, etc.) is a few hundreds milliseconds. In case the calculation time is doubled because of an increasing amount of triangles in a mesh the user will realise this especially for the initial calculation. In order to realise a pseudo adaptive behaviour addressing this problem the user can choose between three different presentation modes for an object.

- Coarse, low calculation time, low visual correctness
- Normal, medium calculation time, medium visual correctness
- Fine, highest calculation time, highest visual correctness

In this way it is up to the user to decide whether to have a high quality presentation but therefore a longer initial delay time and a slower interaction response or a low quality presentation and a short initial delay time and a fast interaction response.

7.4 Interpretation of the Streaming Data

The user requests a VRML file and starts therefore the client application. As an initial step, the object with the most visual presence is sent to the VRML client together with the information of the remaining objects (meta data). The data for the initial object are parsed and displayed on the mobile device client screen. In the next steps the remaining objects are post-loaded in the same way as the initial object. Exceptions are objects which are not visible on the mobile device client screen. For these objects, the information of the event for the post load are sent and stored in the “Event List”. The event is needed because the VRML proxy is unable to push data to the client (HTTP-based communication). Figure 7.2 illustrates the sequence performed to receive the 3D Virtual Reality data on the mobile device client.

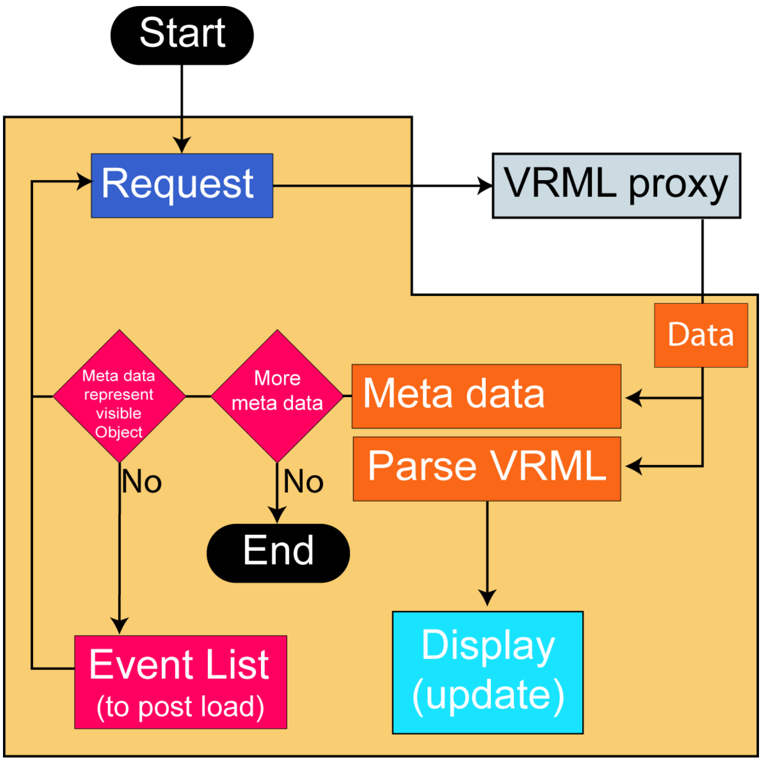


Figure 7.2: Request loop for interaction and event for post load

7.5 Bandwidth Detection

Currently, there are no libraries or functionalities available to directly receive the actual bandwidth information of the connection over the mobile telecommunication system. The physical parameters of the connection and even the type of connection (CSD, GPRS, etc.) are hidden by the operating system.

The only way to obtain information of the bandwidth is trying to measure it. Measuring the real bandwidth of the underlying connection with a Java application is **not** possible. Figure 7.3 shows the sequence diagram performed to obtain a system reaction time.

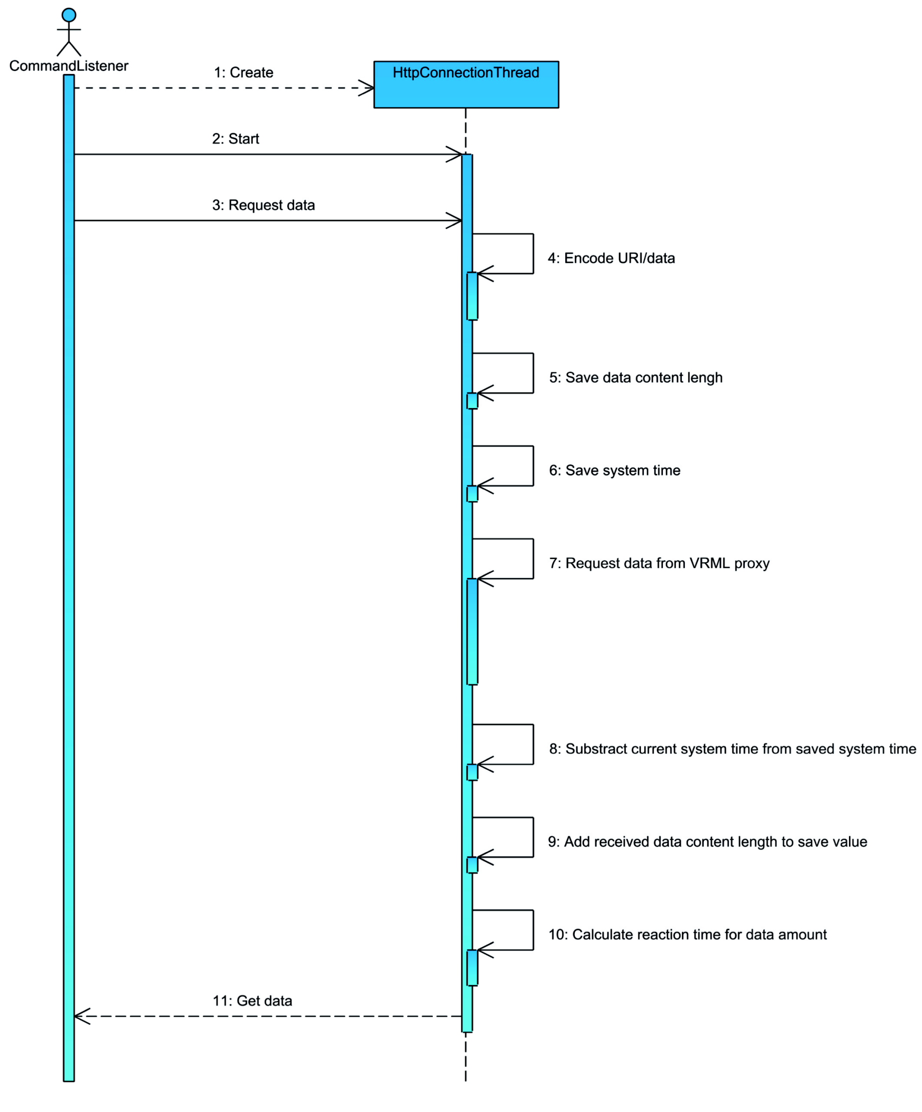


Figure 7.3: Sequence diagram of bandwidth detection

In order to measure the bandwidth one has to evaluate the time and the amount of data transmitted by a request response pair. In case of a synchronous data connection the bandwidth is the amount of data divided by the time. However, in modern mobile telecommunication systems the data connections are not synchronous.

Additionally, the bandwidth is not constant between the request and response because the communication takes place over two different networks: The mobile communication network and the Internet (TCP/IP based network). Thus, the bandwidth is assembled by two different bandwidths. Additional delay time is added by the Dial-In server and the WAP gateway. Furthermore, it is possible to receive the amount of transferred content data but it is impossible to know how much data in total (content data plus control data) were transferred in the request and response, respectively.

As consequence:

With the available information it is impossible to calculate a correct bandwidth. However, for the application it is more important to know an approximate reaction time of the whole system for a certain set of data. The reaction time includes the different bandwidth of both systems as well as the delay times added by the Dial-In server and the WAP gateway. This information can be obtained by the Java application.

7.6 Module Description

The two main components of the mobile device client are the Control module and the Parser module. The Control module performs the user interaction with the application and the VRML proxy. The Parser module generates a software object representation out of the received VRML (X3D) content. Figure 7.4 shows a simplified architecture overview with the additional components needed to present the 3D Virtual Reality data.

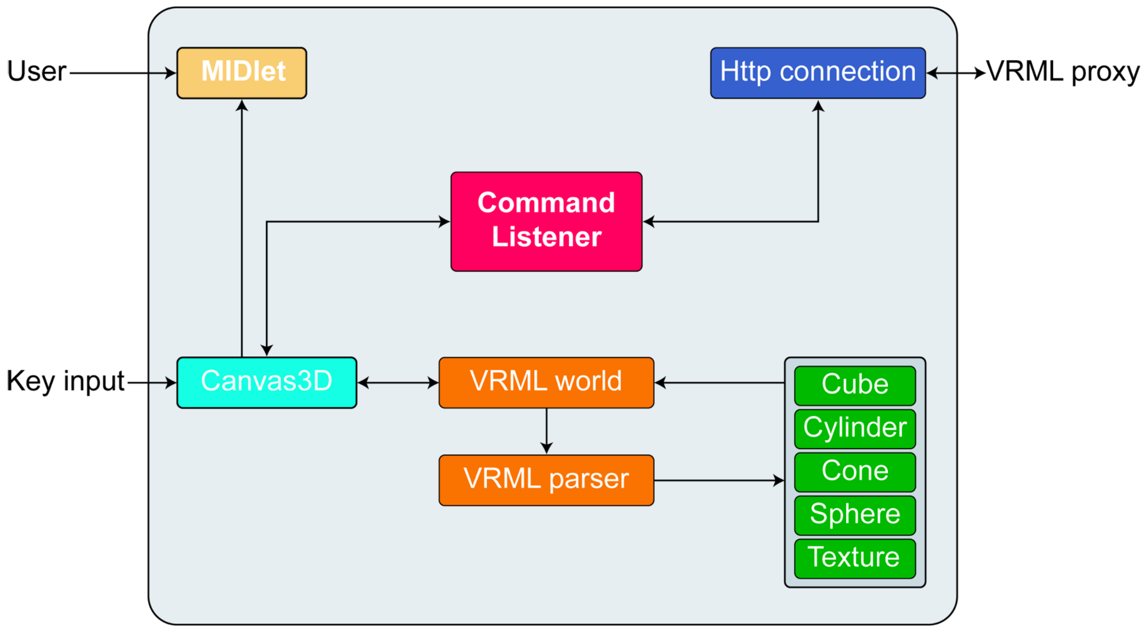


Figure 7.4: Architecture overview of mobile device client

7.6.1 Command Listener Module

The Command Listener module controls and performs the interaction between the user, the application and the VRML proxy. Therefore this module is the core control of the client application. Figure 7.5 shows the steps performed in order to request a specific VRML file from a Web server and to display the data on the mobile device screen.

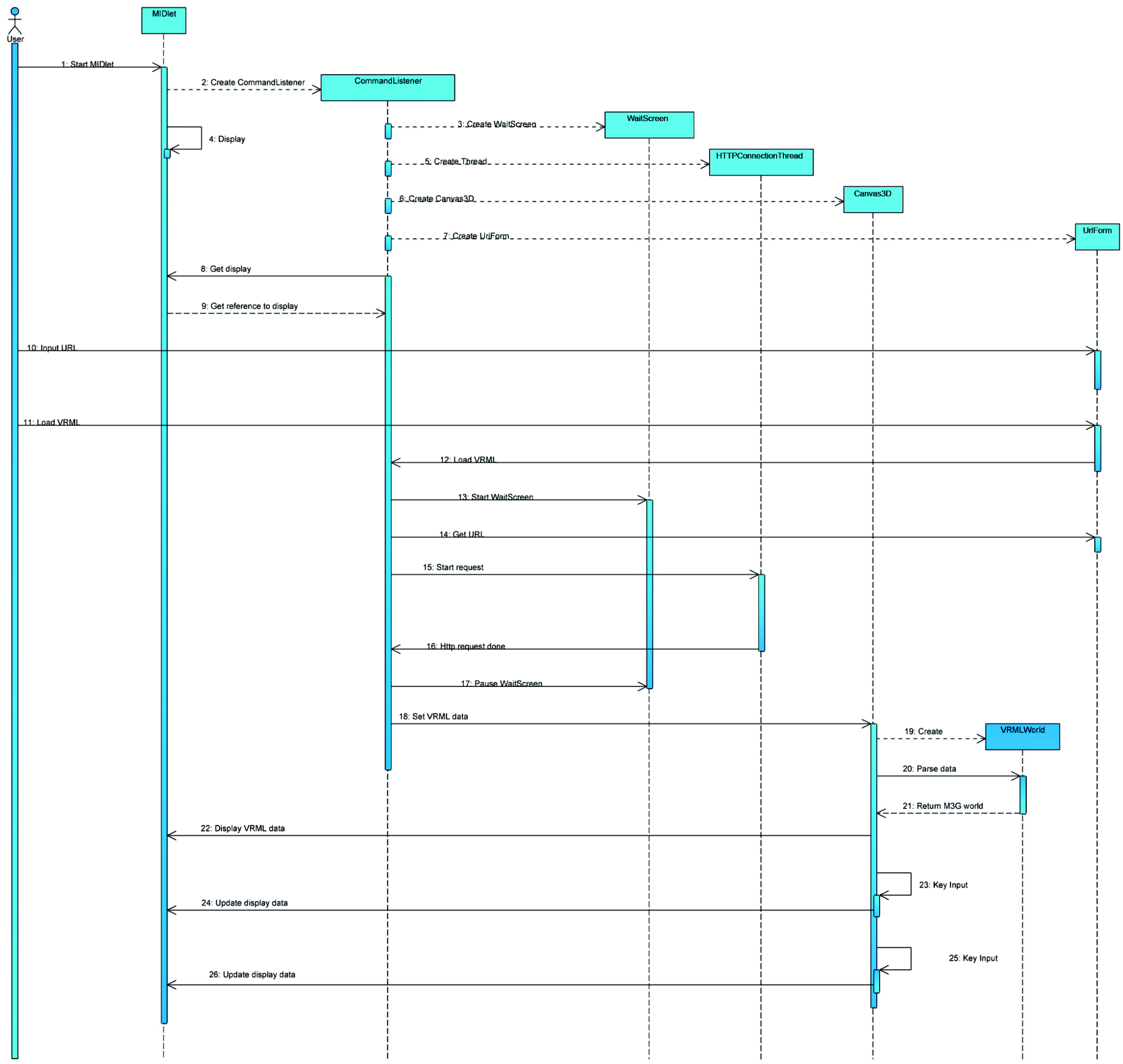


Figure 7.5: Sequence diagram of mobile device client

7.6.2 Parser Module

The parser module is a J2ME implementation of a VRML (X3D) parser which handles a subset (see Table 7.1) of the overall VRML (X3D) nodes. The module reads the content of the VRML proxy response and checks for specific key words indicating the beginning of a definition of a nodes. Figure 7.6 illustrates the steps which are preformed in a sequence in order to detection the nodes [JER04, KOS04, ALLS05].

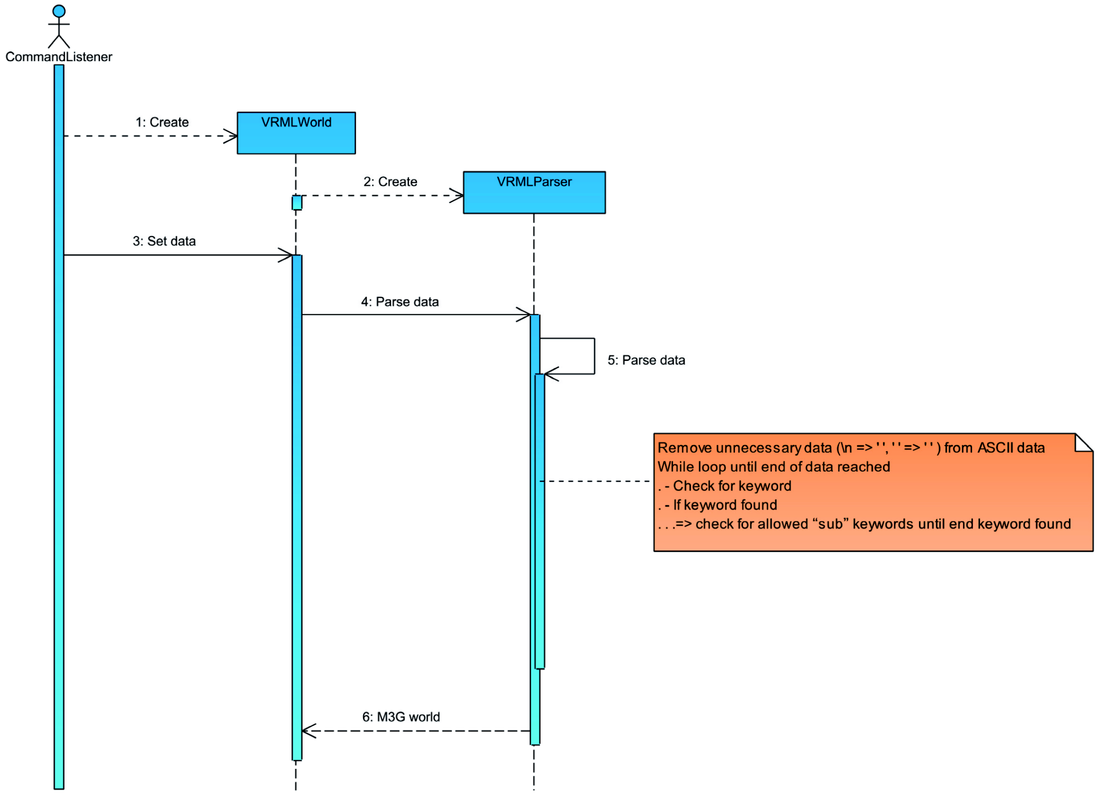


Figure 7.6: Sequence diagram of parser module

In the case that the beginning of a VRML node is detected all values available in the content are read and a corresponding software object is generated.

8 Conclusions

This research project enables users with mobile devices to view 3D Virtual Reality data. The available tools for presenting 3D Virtual Reality data are focused on a stronger processor (at least a PDA) and are working only on specific operating systems (e.g. Pocket Cortona from Parallel Graphics).
The previously discussed work examined three different topics of the system: The network component of modern mobile telecommunication system, the introduced VRML proxy and the mobile device client.

Mobile communication system:

A standard GSM connection with 9,6 KB/s (14,6 KB/s) has a too low bandwidth in order to transmit Virtual Reality data in real time. However, the high latency time (> 2 sec.) has a significant negative influence. It makes it impossible to perform a real time interaction between client and VRML proxy. The bandwidth available in the GSM network increases with the updates to GPRS and HSCSD data connection. Nevertheless, the latency time in the overall system is also high.
The modern mobile telecommunication system (UMTS) offers a significantly higher bandwidth than GSM systems. Additionally, the UMTS system (WCDMA) is a low latency system (< 700 ms). The newly introduced Quality of Service (QoS) allows to obtain defined and guaranteed response times in the system. These QoS features make it possible to realise real time applications.

The VRML proxy component:

With the help of the VRML proxy concept developed in this research it is possible to optimise 3D Virtual Reality data per mobile devices. Additionally, user preferences and settings can be taken into account. Thus, existing 3D Virtual Reality files can be reused without the need to reduce content and to optimise these files for different types of mobile devices. Content providers do not have to care about different mobile devices and do not have to provide an optimised file version for every device.
The important issue concerning this optimisation is the fact that it takes place before the Virtual Reality data is transferred over the mobile telecommunication network. Thus, the optimisation reduces the amount of data which has to be transferred over the mobile telecommunication network.
An additional feature which can be realised by the VRML proxy component is the transformation of any Virtual Reality data format into VRML (X3D) theoretically. This is possible if a Java3D loader for the specific 3D Virtual Reality data format is available (or it has to be implemented). Thus the requested format first can be converted into Java3D at the VRML proxy. The second step performed by the VRML proxy is to output this Virtual Reality data in VRML, optimised for the corresponding mobile device.
The location of a VRML proxy can be either within the network of the content provider or at the service provider. The best location for the VRML proxy is within the network of the service provider because it can be assured that all information (e.g. header) is transferred and not manipulated by any other ”transparent” proxy. Another reason is that a number of mobile devices establish their connection over a WAP gateway to make a HTTP request. In this scenario the best performance can be achieved if the

gateway and the VRML proxy are “close” to each other.
In order to use the highest possible amount of functionality the mobile device must to be able to communicate with the VRML proxy. If this is not the case a subset of the functionality can be used by redirecting requests for 3D Virtual Reality files to the VRML proxy. In that case the functionality is reduced to a “transparent” proxy which reduces content.

The mobile device client:

The improvement of the hardware of mobile devices is increasing rapidly especially in the sector of mobile phones and smartphones. In between the beginning of the year 2005 and 2006 the average main memory of mobile phones has developed from about 0.5 MB to around 20 MB. The average storage capacity has increased from 32 MB often not pluggable to 512 MB pluggable via a flash card. At the beginning of 2006 the newest high-end mobile phones offered up to 80 MB main memory and 1 GB storage memory.
Mobile devices with a build-in 3D screen are available on the Asian market only. However, the newest development of mobile phones offers 3D hardware acceleration. Additionally, the screen size of mobile phones and smartphones is increasing and the newest models have a screen size of up to 480 by 640 pixels.
The apprehension that Java may be to slow and not capable to handle the 3D presentation of the 3D Virtual Reality data has shown causeless. The reasons for that are on one hand the increasing processing power of the mobile devices and on the other hand the Java hardware support realised by the manufacturer. More and more manufacturers realised the potential of Java, especially concerning mobile gaming, and thus are optimising their hardware and software and adding hardware accelerated 3D Java support to the devices. New Java features such as the build in Remote Method Invocation (RMI) support makes it possible to realise “thin clients”. This fact is an important point in the increasingly mobility-driven world.
The research has shown that it is possible with actual phones to realise 3D Virtual Reality viewers with both monoscopic and stereoscopic presentations. The stereoscopic presentation has to be performed with additional external devices due to the lack of built-in 3D screens. Unfortunately, this limits the user in his mobility. Nevertheless this presentation type has shown 3D scenes as an impressive and realistic 3D Virtual Reality.
Monoscopic presentation cannot be that impressive and realistic because the depth information is missing. Even though the monoscopic presentation provides a clear and convenient presentation of 3D objects. This pseudo 3D presentation transports additional information to the user. With the help of interaction the user can examine and study 3D objects from any perspective. Thus the user can gain additional information and knowledge.

The goal of the research was to allow users with mobile devices to be able to view and interact with 3D scenes such as users on personal computers. By means of the VRML proxy the 3D Virtual Reality data is optimised for transmission and the mobile devices. Thus, it is possible to present and to interact with 3D Virtual Reality data. Although the mobile devices cannot establish a 3D Virtual Reality presentation with that much feature as on a personal computer this research has shown that the principle abilities are available.

9 References

9.1 Books, Thesis and Project Work

[ALLS05] Visualisation of 3D data transmitted by Mobile Phone
ARMAND Damien, LEGENDRE Arnaud, LISIMAQUE Vincent, SCHWARTZ Antoine
Project 15, LSP 2005

[BRU98] Thinking in Java
ECKEL Bruce
Prencice HALL PTR, 1998

[ED01] Mobiles Internet
EREN Evren, DETKEN Kai-Oliver
Addison-Wesley, München 2001

[FDFH90] Computer Graphics, Principle and Practice Second Edition
FOLEY James D., DAM Andries van, FEINER Steven K., HUGHES John F.
Addison-Wesley 1990

[FDFH94] Grundlagen der Computergrafik Einführung, Konzeption, Methoden
FOLEY James, VAN DAM Andries, FEINER Steven , HUGHES John,
PHILLIPS Richard
Addison-Wesley 1994

[HEI00] Understanding WAP
HEIJDEN van der M.
Artech House, 2000

[HH02] UMTS and mobile computing
HUBER Alexander, HUBER Josef
Artech House, London 2002

[HK01] Essential WAP for Web Professionals
HOUGLAND Damon, ZAFAR Khurran
Prentice Hall,2001

[HT01] WCDMA for UMTS
HOLMA Harri, TOSKALA Antti
John Wiley & Sons Ltd. , Chichester 2001

[JER04] Virtual Reality – Visualization Module for Mobile Devices:
Communication & Rendering
JEREZ Carlos
Master Thesis University of Applied Sciences Offenburg, 2004

[KOS04] Virtual Reality – Visualization Module for Mobile Devices:
User Interface & Model Behavior
KOSHELKINA Olga
Master Thesis University of Applied Sciences Offenburg, 2004

[LSP] Laboratoire des Systèmes Photoniques (LSP)
Head Mounted Devices
(Prototype developed by Project 14, 2005 ROGER Thomas, LAUSTRIAT Nicolas, RAULOT Victorien, CARDINALI Vanessa)
Stereoscopic Video Projection System

[MC97] VRML2 in 21 deays

MARRIN Chris, CAMPBELL Bruce
sams.net, Indianapolis 1997

[MEN02] The Streaming Media Handbook
MENIN Eyal
Prentice Hall, August 2002

[RAU93] Algorithmen in der Computergrafik
RAUBER Thomas
B.G. Teubner, Stuttgart 1993

[RCRR97] Light Night VRML2 with Java
ROEHL Bernie, COUCH Justin, REED-BALLREICH Cindy, ROHALY Tim, BROWN Geoff
Ziff-Davis Press, Emeryvill 1997

[RWO95] An Introduction to GSM
REDL Siegmund, WEBER Matthias, OLIPHANT Malcolm
Artech House, London 1995

[SW63] Mathematical Theory of Communication
SHANNON Claude, WEAVER Warren
University of Illinois Press, December 1963

[TL94] 3D-Computergrafische Darstellung
TÖNNIES Klaus, LEMKE Heinz
Oldenbourg Verlag GmbH, München 1994

[TOP02] Streaming Media Demystified
TOPIC Michael
McGraw-Hill Professional Publishing, May 2002

[WEN04] Modélisation 3D à partir de photograpies non calibrées
WENDLING Thomas
PROJECT DE FIN D'ETUDES, 2004

9.2 Magazines, Journals and Conferences

[BUR04] Multimedia Processor for Handhelds Punches the Graphics, Video Gas Pedal
BURSKEY Dave
ED Online 7718 Electronic Design, 04.12.2004

[CV05] Video Adaptation: Concept, Technologies, and Open Issues
CHANG Shih-Fu, VETRO Anthony
IEEE Proceedings, Vol.93 No. 1, January 2005

[DON02] Mobile phone are next venue for 3D graphics
DONELAN Jenny
Computer Graphics World (http://www.cgw.com), October 2002

[EY05] Advances in Wireless Video Delivery
ETOH Minoru, YOSHIMURA Takeshi
IEEE Proceedings, Vol.93 No. 1, January 2005

[HOL05] 3D Display Systems
HOLLIMAN Nick
Department of Computer Science, University of Durham, February 2, 2005

[MAN97] Wearable computing: A first step toward personal imaging
MANN Steve

IEEE Computer, Feb 1997

[MAN98] WEARABLE COMPUTING as means for PERSONAL EMPOWERMENT
MANN Steve
International Conference on Wearable Computing ICWC-98, Fairfax VA, May 1998
http://wearcomp.org/wearcompdef.html

[PASI] Producing Anaglyphs from Synthetic Images
SANDERS William, MCALLISTER David
Department of Computer Science, North Carolina State University, Raleigh

[SIP04] Special Interest Paper:
Mobile 3D Graphics and Java Applications Development for Sony Ericsson Phones
Sony Ericsson, Novemer 2004

[SJK06] Methods for Displaying Three-Dimensional Images
SON Jung-Young, JAVIDI Bahram, KWACK Kae-Dal
IEEE Proceedings Vol. 94 No. 3, Mach 2006

[SYM04] Java Development on Symbian OS
JONATHAN Product manager Java, Symbian Ltd.
Symbian expo, 5/6 October 2004

[TM02] JPEG2000: Standard for Interactive Imaging
TAUBMAN David, MARCELLIN Michael
IEEE, 1336 Vol 90, Nr. 8 August 2002

[UQOS] Universal Mobile Telecommunications System (UMTS) QoS Concept
UMTS 23.07 version 0.3.0
ETSI http://www.etsi.org, 1999

[WIL01] Playing Handset Game (Accelerated 3D graphics for mobile phones)
WILSON Riachard
Electronics News, 28 May 2001

9.3 World Wide Web

[ABO99] Mobile and Ubiquitous Computing Reading List
ABOWD Gregory
http://c2000.cc.gatech.edu/classes/cs8113c_99_spring/readings/overview.html

[APACHE] HTTP Apache
http://jakarta.apache.org/commons/httpclient

[ARA] ARAIZ Carlos
http://www.java4ever.com/
index.php?section=j2me&project=gzip&menu=main&lang=_en

[BEI04] Virtual Reality: A Short Introduction
BEIER Klaus-Peter
10 February 2004, http://www-vrl.umich.edu/intro/index.html

[BEN] Mobile Phone Resources, Developer Network
http://www.benhui.net

[BOU] Sun's JAVA3D Tutorial
Getting started with JAVA 3D API Version 1.5 (for API 1.1.2)
BOUVIER Dennis

[DCC] Data Compression Comparison
http://compression.ca/act

[DCTC] Data Compression Theories & Comparison
http://www.data-compression.com

[EJCIO] Essential Java Classes, I/O: Reading and Writing
http://java.sun.com/docs/books/tutorial/essential/io/

[EJCT] Essential Java Classes, Threads: Doing Two or More Tasks at Once
http://java.sun.com/docs/books/tutorial/essential/threads/index.html

[EPG] E-Plus Germany
http://www.eplus.de

[GCC] GNU Compiler Collection (C/C++, Fortran, Java)
http://gcc.gnu.org

[GC] Game Canvas
http://developers.sun.com/techtopics/mobility/midp/ttips/gamecanvas/index.html

[GIFS] GIF Specification
http://www.w3.org/Graphics/GIF/spec-gif89a.txt

[GZIP] Gnu ZIP Organisation
http://www.gzip.org

[H3DW] How 3D Display Works, Stereoscopic
http://www.sharp3d.com/technology/how3ddisplayworks

[HCMC] HI CORP Mascot Capsule (M3G library and toolkit used by Sony Ericsson)
http://www.mascotcapsule.com/M3G/e_index.html

[HS3W] How Sharp 3D Works
http://www.sharp3d.com/technology/howsharp3dworks

[IWG03] Introduction to WAP over GPRS
Nokia, 30. April 2003 http://forum.nokia.com

[J3ASE] Java 3D API Specification and Examples
http://www.j3d.org

[J3RE] J2ME 3D Rendering Engine
http://rzr.online.fr/docs/java/00index.htm

[JCP] Java Community Process
http://www.jcp.org

[JLT03] J2ME Latency Test
http://www.admpartners.com/blog/2003/07/latency_test_re_1.html

[JMR] Java Mobile References, Tutorials, Examples
http://java.sun.com/j2me
http://developers.sun.com/techtopics/mobility/midp/articles/midp2network

[JN] J2ME Network
http://www.wirelessdevnet.com/channels/java/features/j2me_http.phtml

[JRS] Java RTP Support
http://java.sun.com/products/java-media/jmf/2.1.1/support-rtp.html

[LCD04] Liquid Crystal Displays
Ref. No. HT549D SHARP CORPORATION. OCT. 2004

[M3G] Mobile 3D Graphics (M3G), JSR-184
http://www.jcp.org/en/jsr/detail?id=184

[MIC96] Virtual Reality
KEITH Michell
16. December 1996, http://ei.cs.vt.edu/~history/Mitchell.VR.html

[MPFB] Mobile Phone Features and J2ME Benchmark
http://www.club-java.com/TastePhone/J2ME/MIDP_Benchmark.jsp

[MRMI] J2ME RMI
http://www.symbian.com/developer/techlib/papers/
PJAE_MIDP/PJAE_MIDP_2.pdf

[NDF] Nokia Developer Forum (Technical Papers)
http://www.forum.nokia.com

[O2G] O2 Germany
http://www.o2-online.de

[PGPC] Parallel Graphics, Cortona and Pocket Cortona
http://www.parallelgraphics.com/products/cortona
http://www.parallelgraphics.com/products/cortonace

[PNG] Portable Network Graphics
http://www.w3.org/Graphics/PNG

[RFC1952] GZIP RFC
http://www.ietf.org/rfc/rfc1952.txt?number=1952

[RFC2616] HTTP 1.1
http://www.ietf.org/rfc/rfc2616.txt?number=2616

[RMI] RMI
http://java.sun.com/developer/technicalArticles/RMI/rmi_corba

[RNC] RealNetworks Customer Support
http://service.real.com/help/library

[RSTP] RTSP (Rel-time Streaming Protocol)
http://www.rtsp.org

[SCE] Stereographics Cristal Eyes, Shutter glasses & Tracking system
Owned by the University of Applied Science

[SEDF] Sony Ericsson Developer Forum (Technical Papers)
http://developer.sonyericsson.com

[TG] T-Mobile, German
http://www.t-mobile.de

[VLC] VRML Loader, CyperVRML97
http://www.cybergarage.org/vrml/cv97/cv97java/index.html

[VG] Vodafone Germany
http://www.vodafone.de

[VRML] VRML97 Specification, Node description
http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/part1/nodesRef.html

[W3C] web3D Consortium
http://www.web3d.org

[WAP02] Wireless Application Protocol (WAP 2.0), Technical White Paper
WAP Forum, http://www.wapformun.org, Januar 2002

[WHE03] Lenticular Stereogram Maker
WHENER Charles
http://wehner.org/tools/len

[WMDD] Windows Mobile DirectX and Direct3D
http://msdn2.microsoft.com/en-US/library/ms172504.aspx

10 Appendix

10.1 List of Tables

10.2 List of Figures

10.3 List of mobile client supported/unsupported VRML nodes

Object	***Status in Project***
Anchor	not supported
Appearance	supported
AudioClip	not supported
Background	supported
Billboard	not supported
Box	supported
Collision	not supported
Color	supported
ColorInterpolator	not supported
Cone	supported
Coordinate	not supported
CoordinateInterpolator	not supported
Cylinder	supported
CylinderSensor	not supported
DirectionalLight	supported
ElevationGrid	not supported
Extrusion	not supported
Fog	not supported
FontStyle	not supported
Group	not supported
ImageTexture	supported
IndexedFaceSet	not supported
IndexedLineSet	not supported
Inline	not supported
LOD	not supported
Material	supported
MovieTexture	not supported
NavigationInfo	not supported
Normal	not supported
NormalInterpolator	not supported
OrientationInterpolator	not supported
PixelTexture	not supported
PlaneSensor	not supported
PointLight	not supported
PointSet	not supported
PositionInterpolator	not supported
ProximitySensor	not supported
ScalarInterpolator	not supported
Script	not supported

Object	***Status in Project***
Shape	supported
Sound	not supported
Sphere	supported
SphereSensor	not supported
SpotLight	not supported
Switch	not supported
Text	not supported
TextureCoordinate	not supported
TextureTransform	not supported
TimeSensor	not supported
TouchSensor	not supported
Transform	supported
Viewpoint	supported
VisibilitySensor	not supported
WorldInfo	not supported