

“Window into a virtual world” screen concept

SEP Final

17. 10. 2008 Nicolas Heuser
(heuser@in.tum.de)

Overview

- Problem
- Fundamentals / Definitions
- Offaxis Projection
- Implementation details
- Future work
- Q&A
- Demo

Problem

Problem

- Find a way to easily extend 3D applications to support CAVE and Powerwall environments

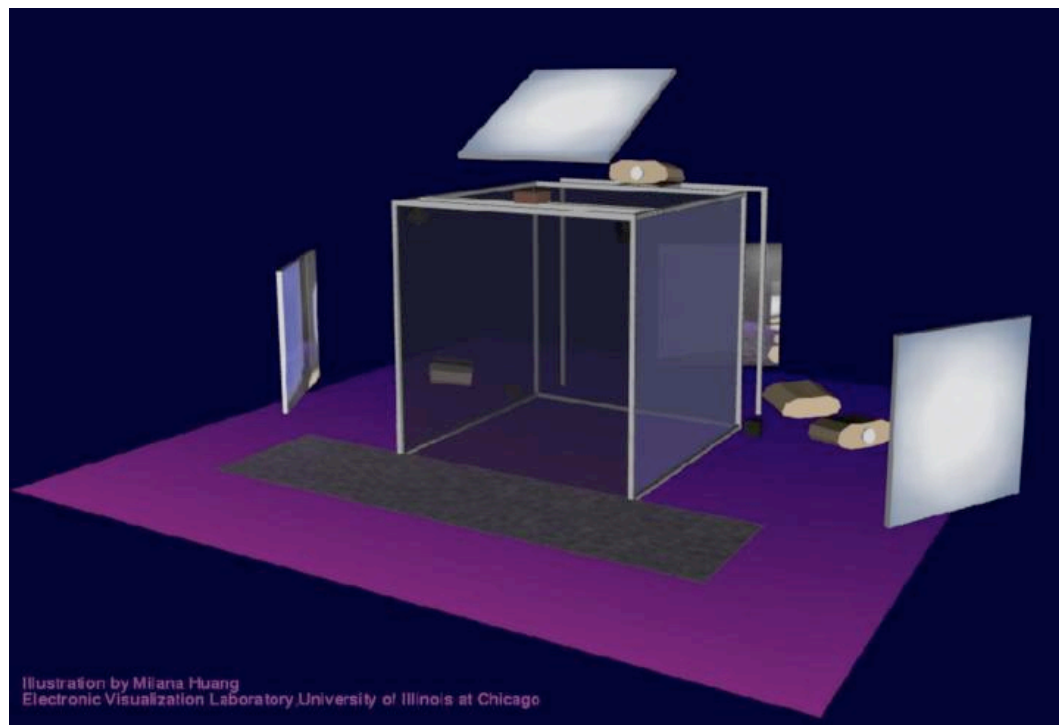
Problem / Requirements

- Find a way to easily extend 3D applications to support CAVE and Powerwall environments
 - support for various display configurations
 - support “realtime” rendering
 - easy setup

Problem / Requirements

- Find a way to easily extend 3D applications to support CAVE and Powerwall environments
 - support for various display configurations
 - support “realtime” rendering
 - easy setup
 - Ubitrack component

Fundamentals - CAVE environment



- arbitrary size and number of sides
- walls don't have to be perpendicular

<http://archive.ncsa.uiuc.edu/Cyberia/VETopLevels/VR.Systems.html>

Fundamentals - Powerwall



<http://www.comp.leeds.ac.uk/vvr/>

- arbitrary size and numbers of displays
- support for headtracked and non-headtracked viewers

Solution: “Window into a virtual world” screen concept



- arbitrary location and orientation of the screen
- independent to viewer position
- calculated in “real-time”

Real world example



Real world example



Real world example



Real world example

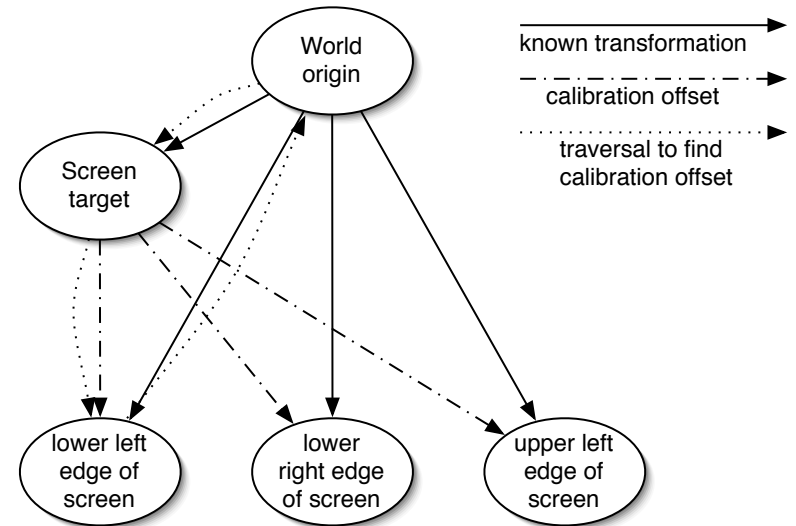


Virtual counterpart



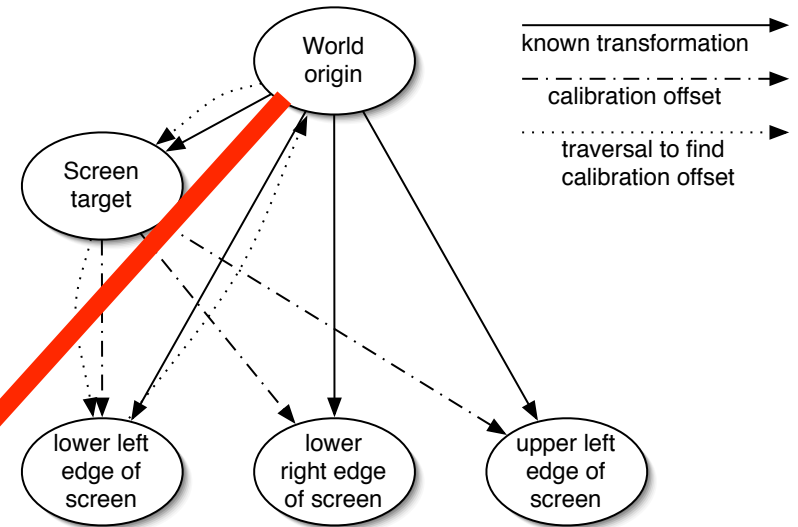
Registration and calibration

- match virtual and real world coordinate systems
- find the relative offset from marker target to screen corners



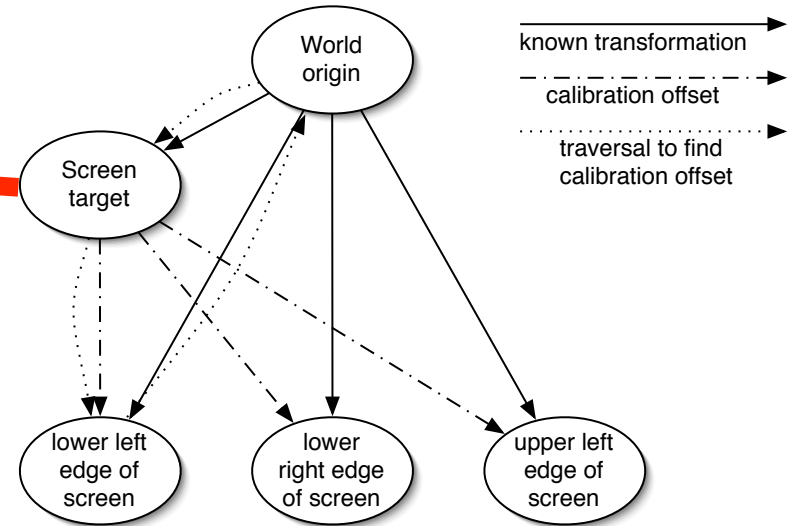
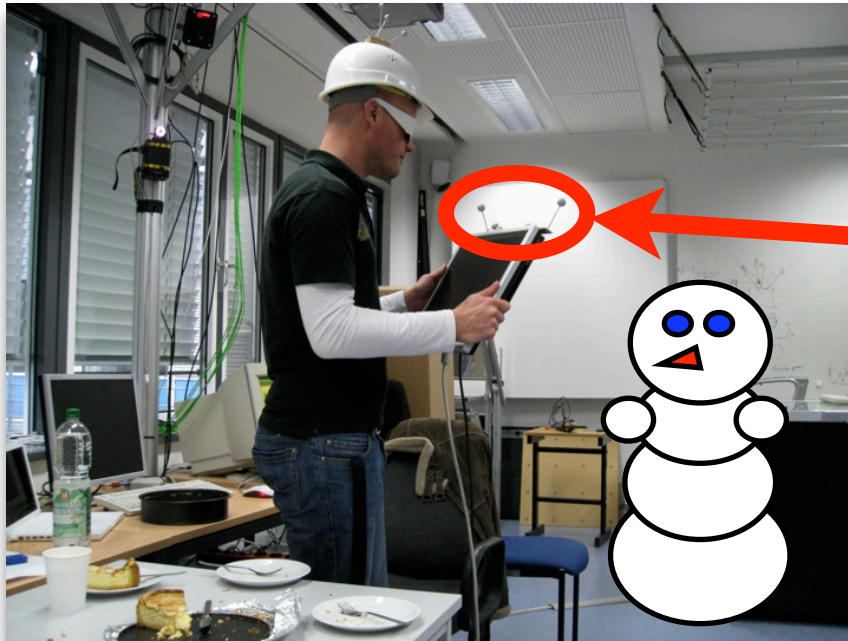
Registration and calibration

- match virtual and real world coordinate systems
- find the relative offset from marker target to screen corners



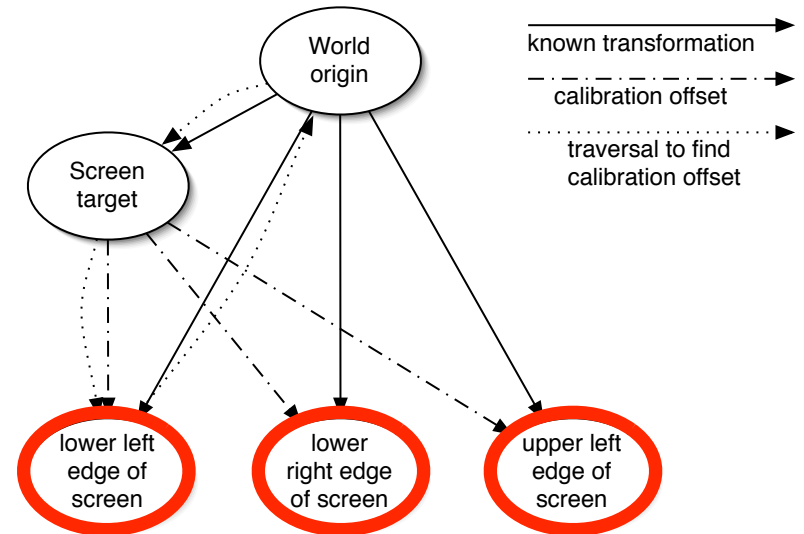
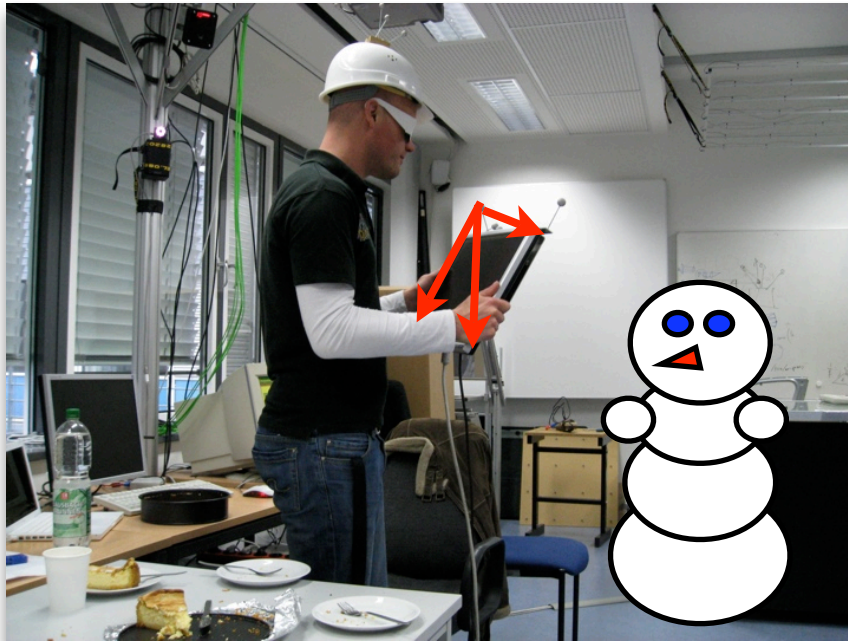
Registration and calibration

- match virtual and real world coordinate systems
- find the relative offset from marker target to screen corners

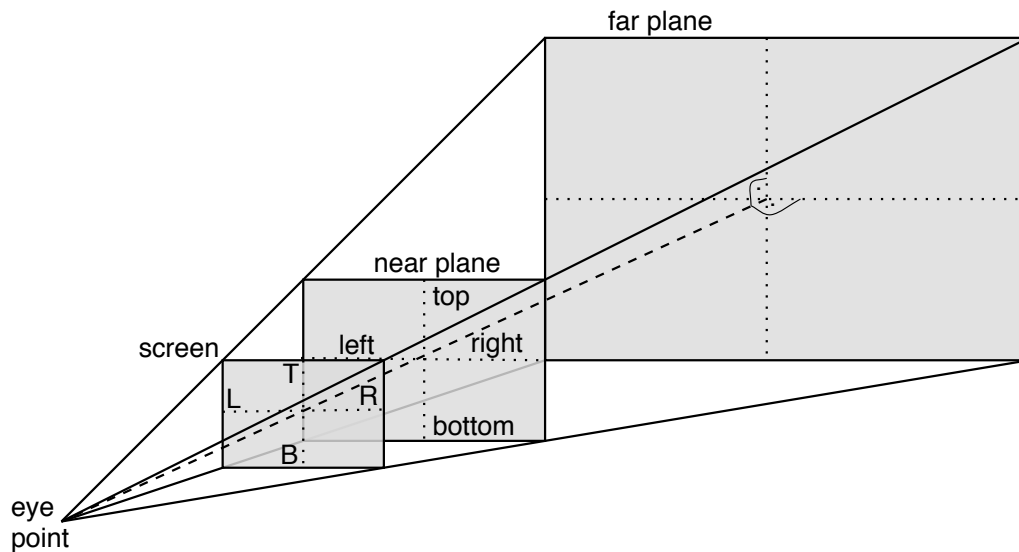


Registration and calibration

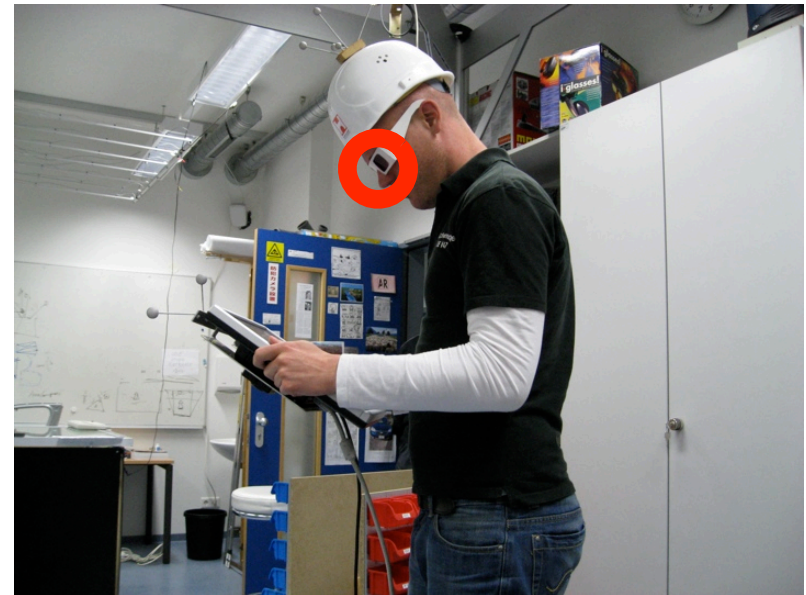
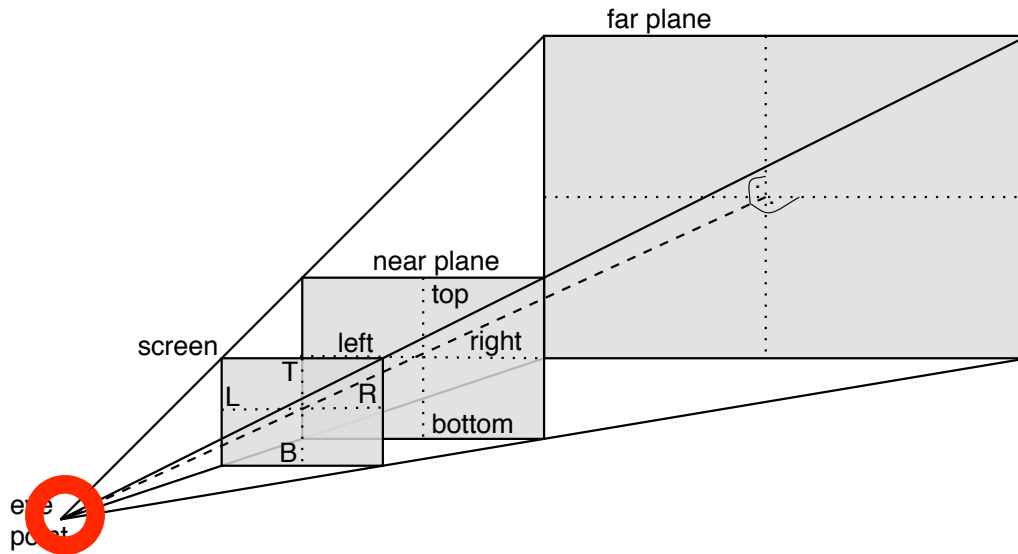
- match virtual and real world coordinate systems
- find the relative offset from marker target to screen corners



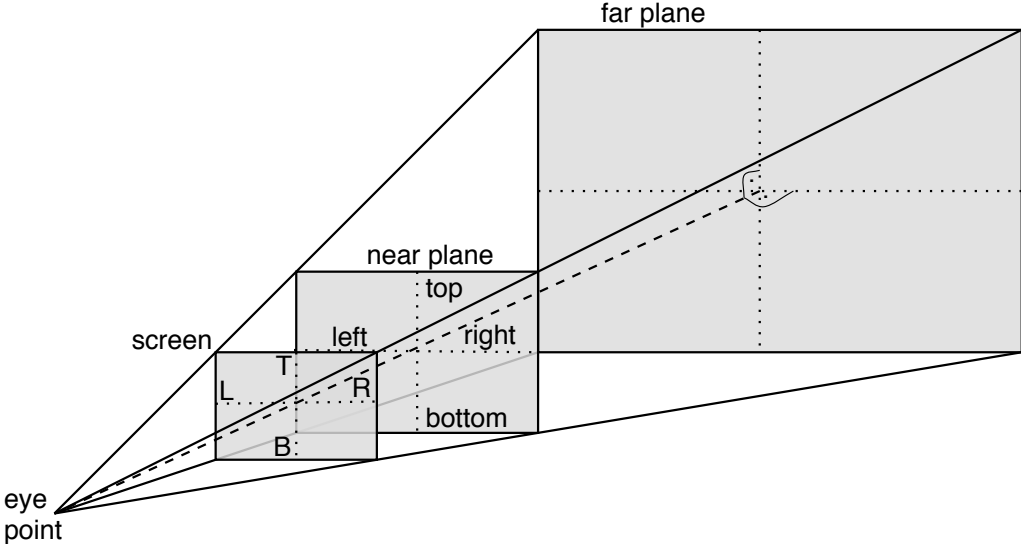
Definitions - Eye point



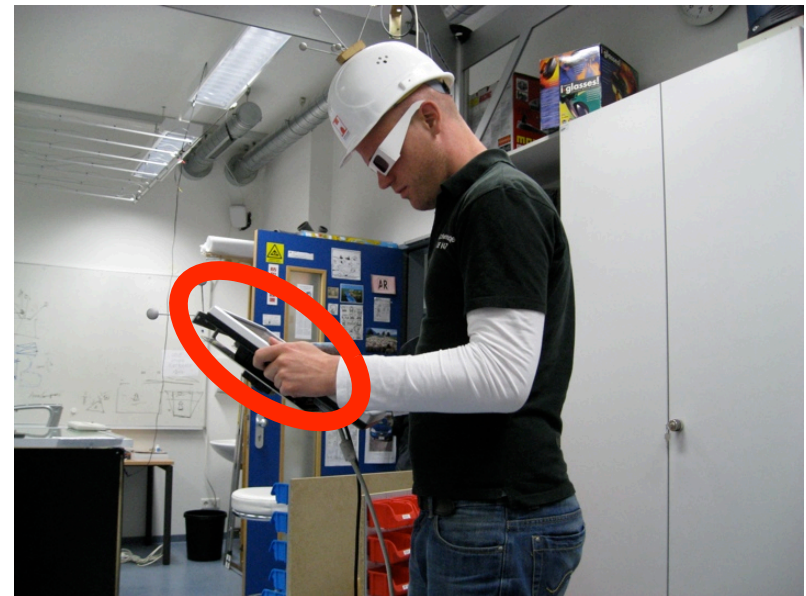
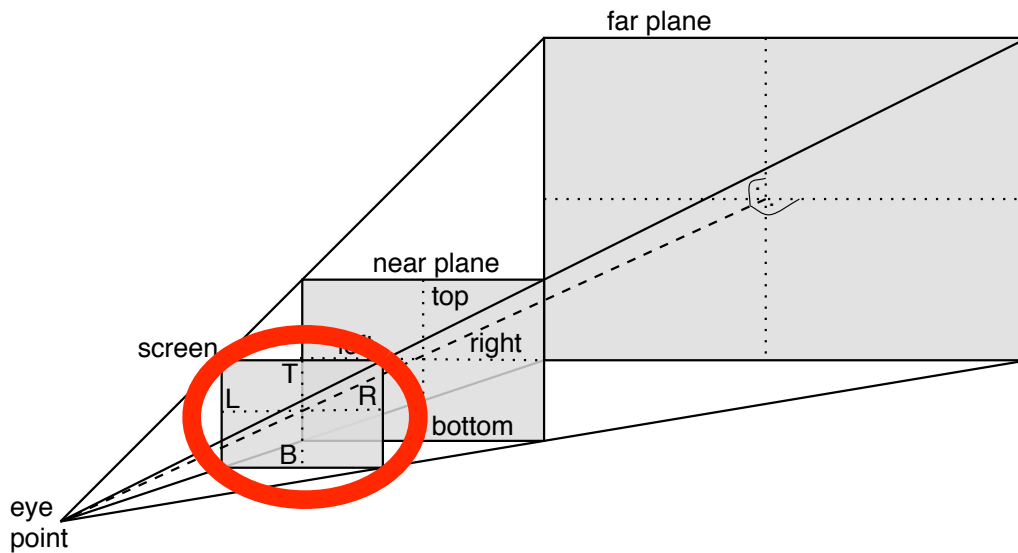
Definitions - Eye point



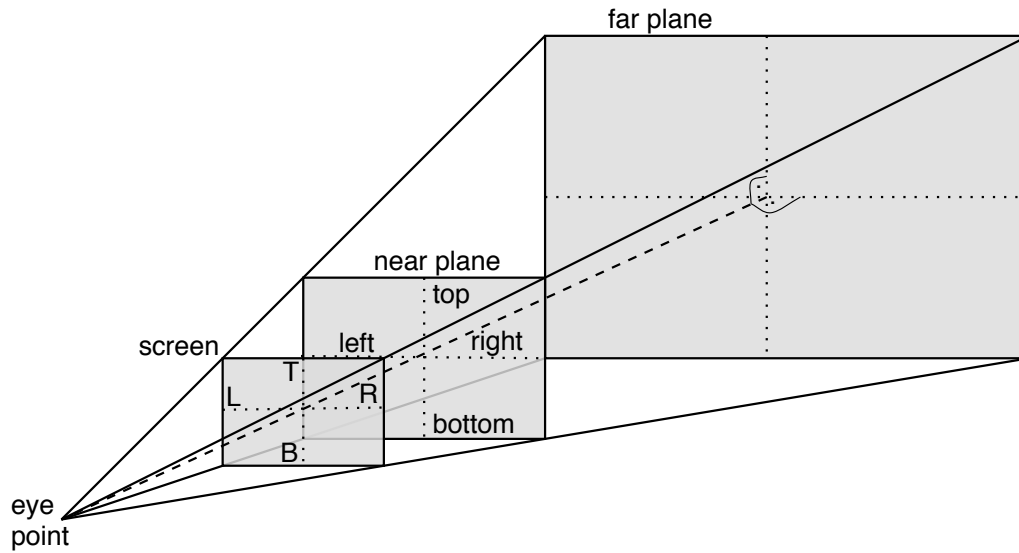
Definitions - Screen plane



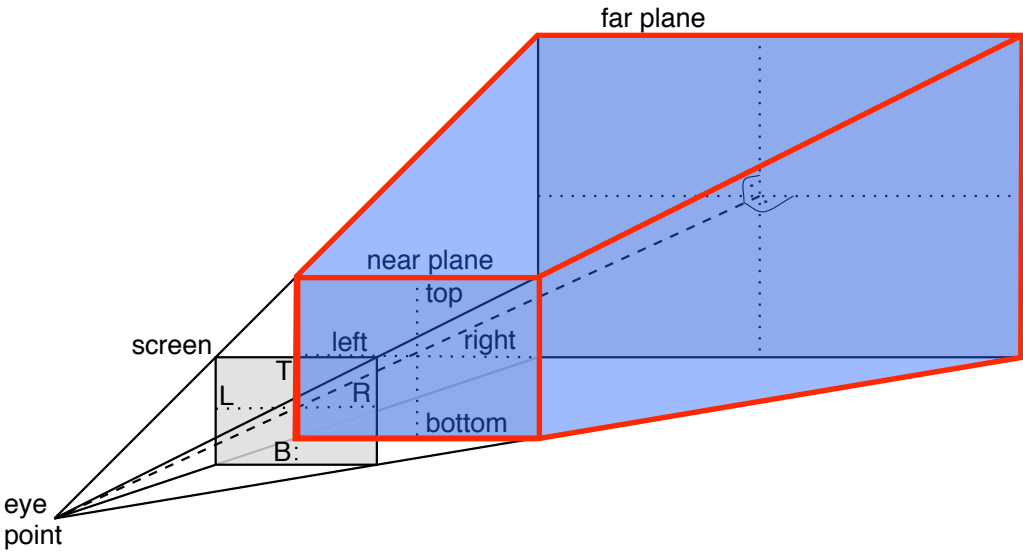
Definitions - Screen plane



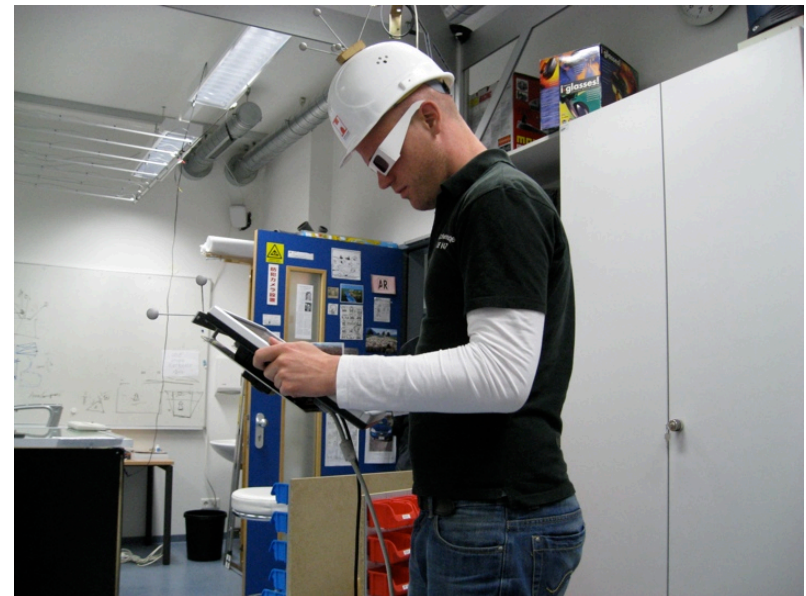
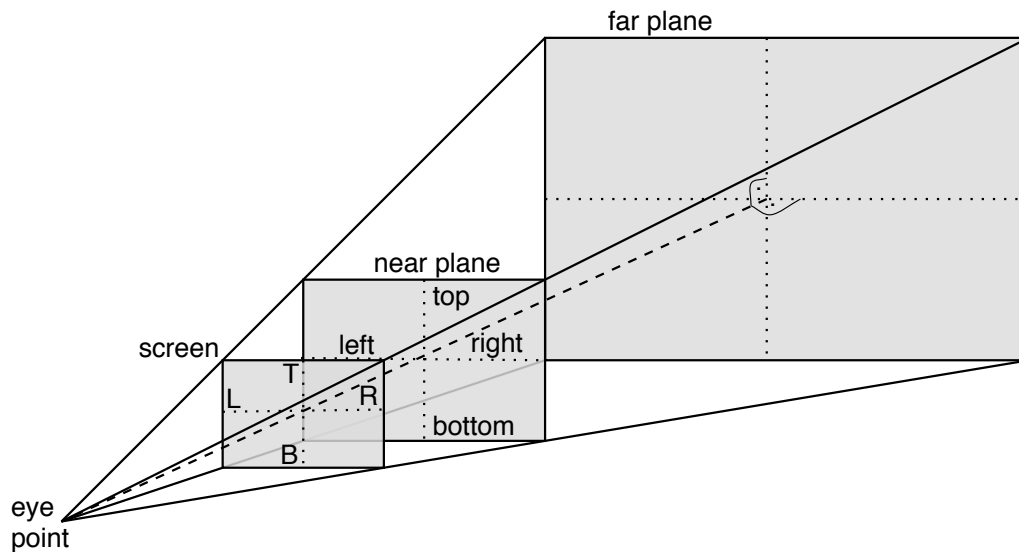
Definitions - View frustum



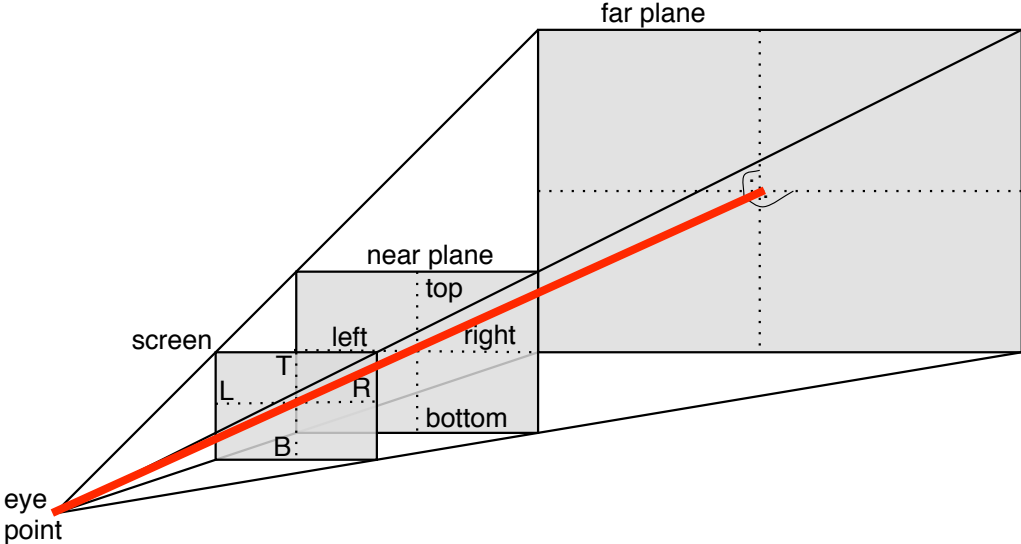
Definitions - View frustum



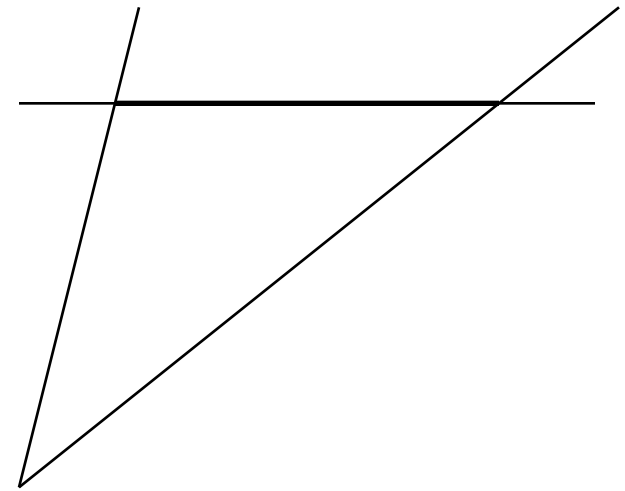
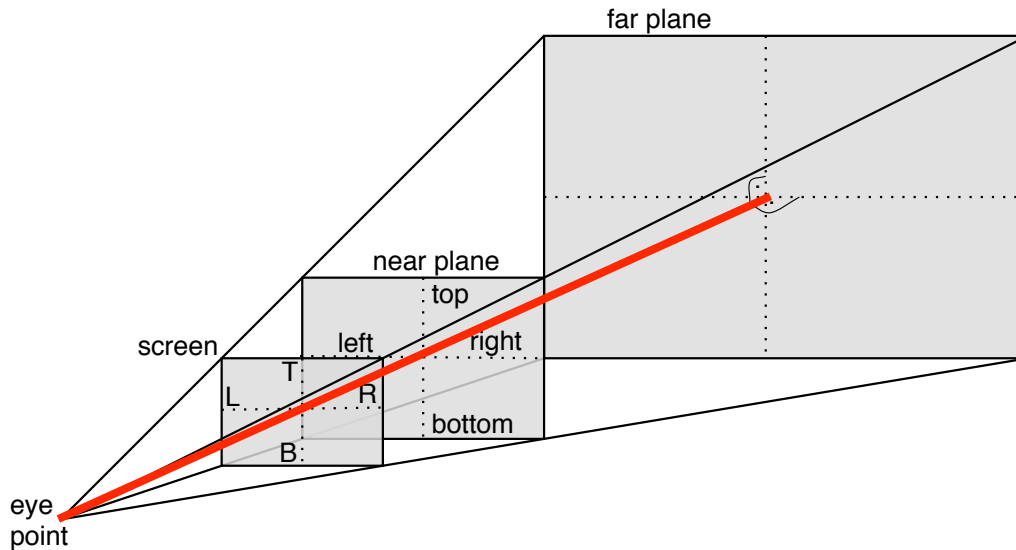
Definitions - View vector



Definitions - View vector

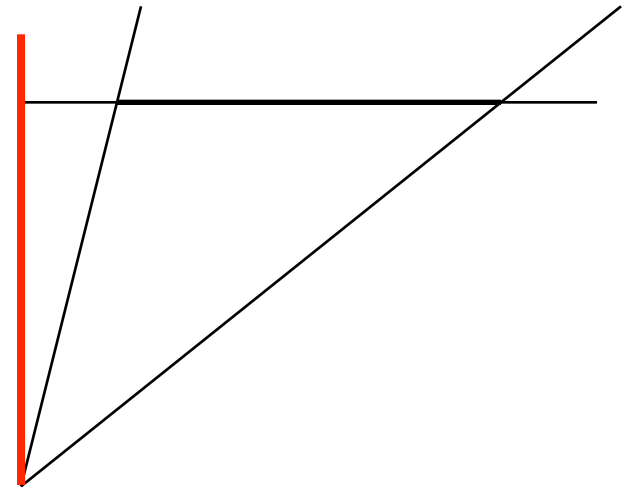
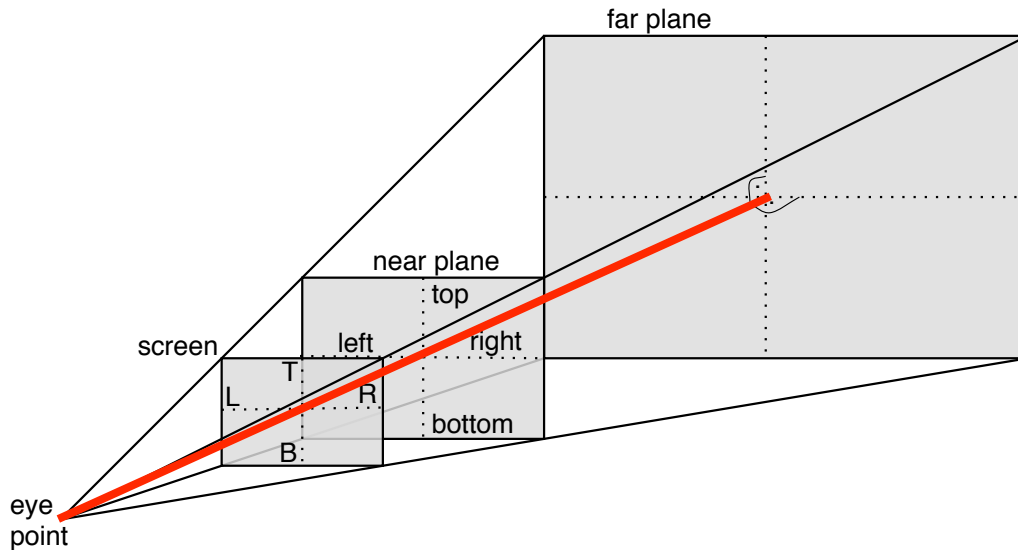


Definitions - View vector



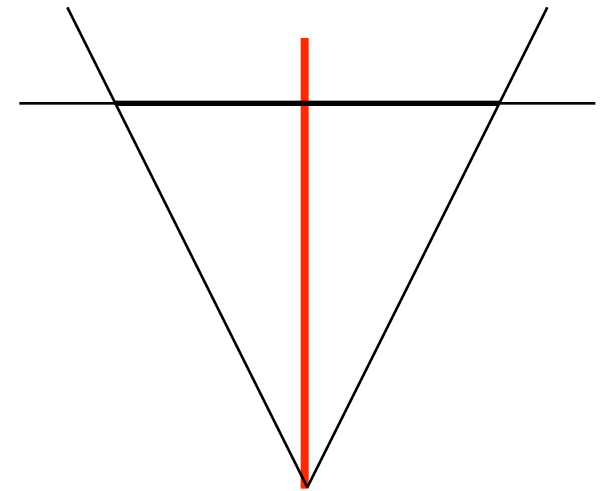
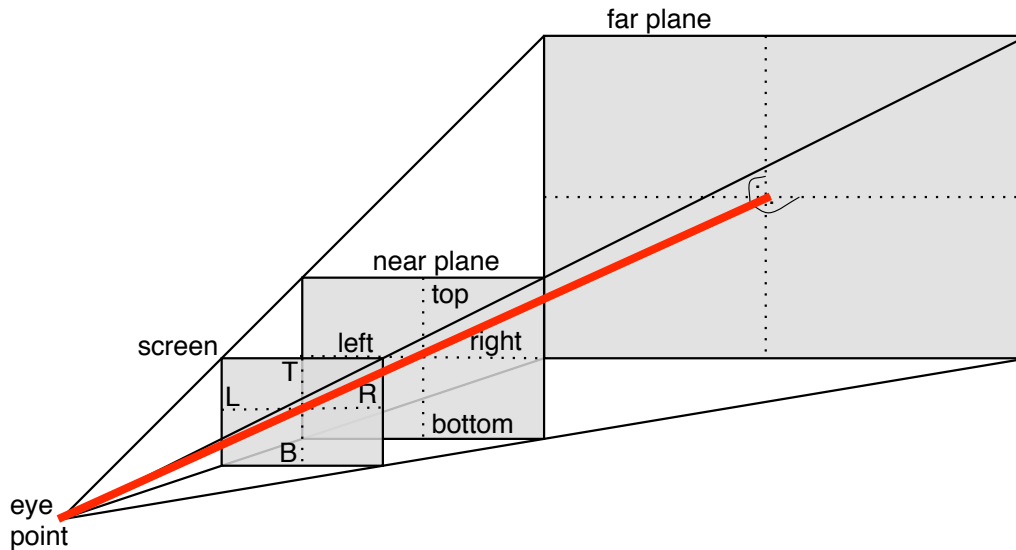
View vector is always perpendicular to the screen plane

Definitions - View vector



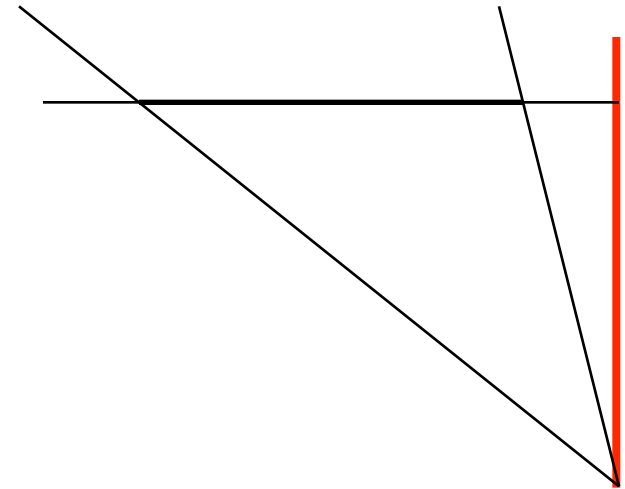
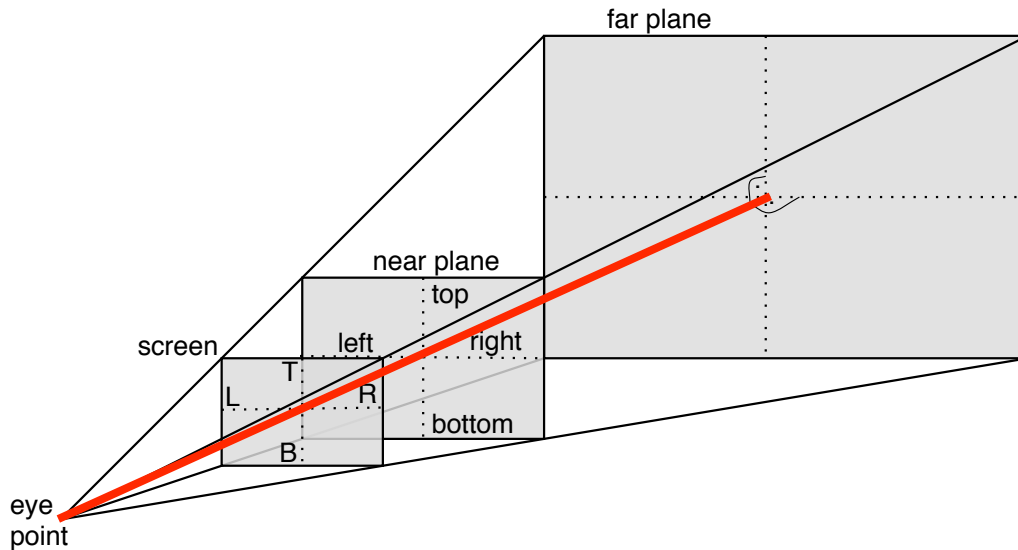
View vector is always perpendicular to the screen plane

Definitions - View vector



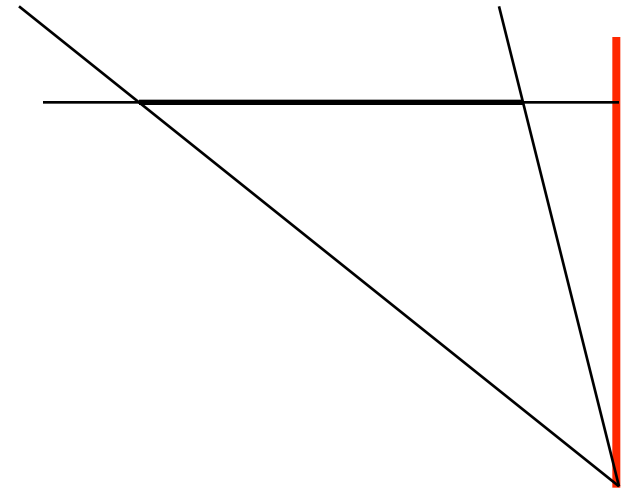
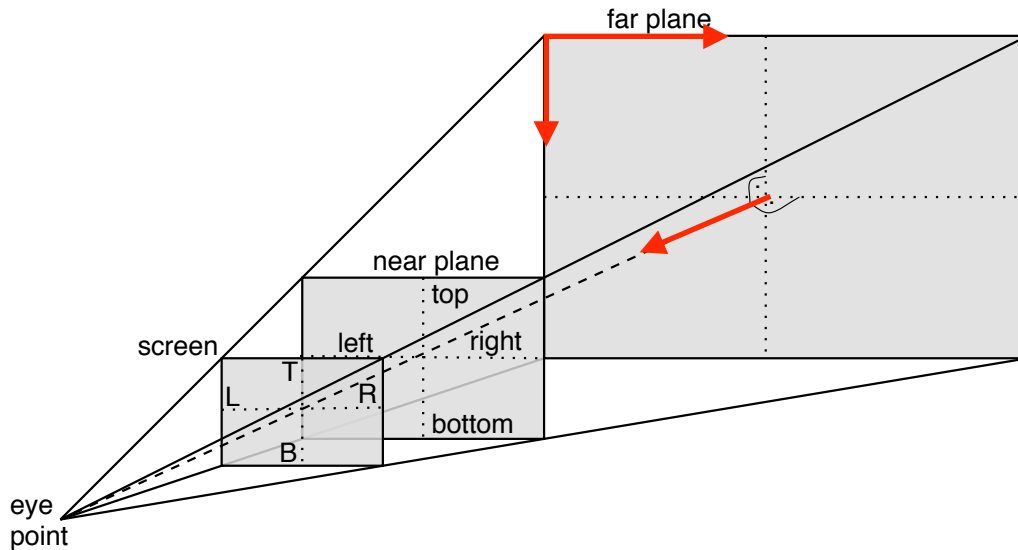
View vector is always perpendicular to the screen plane

Definitions - View vector



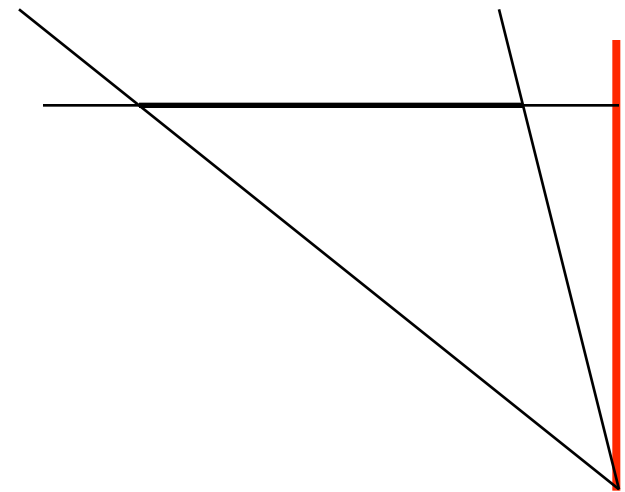
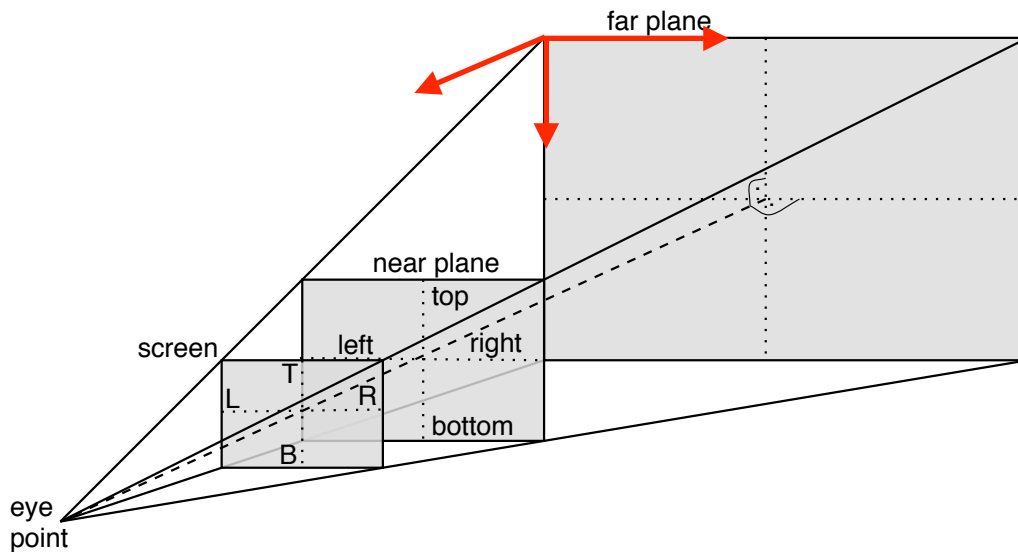
View vector is always perpendicular to the screen plane

Definitions - View vector



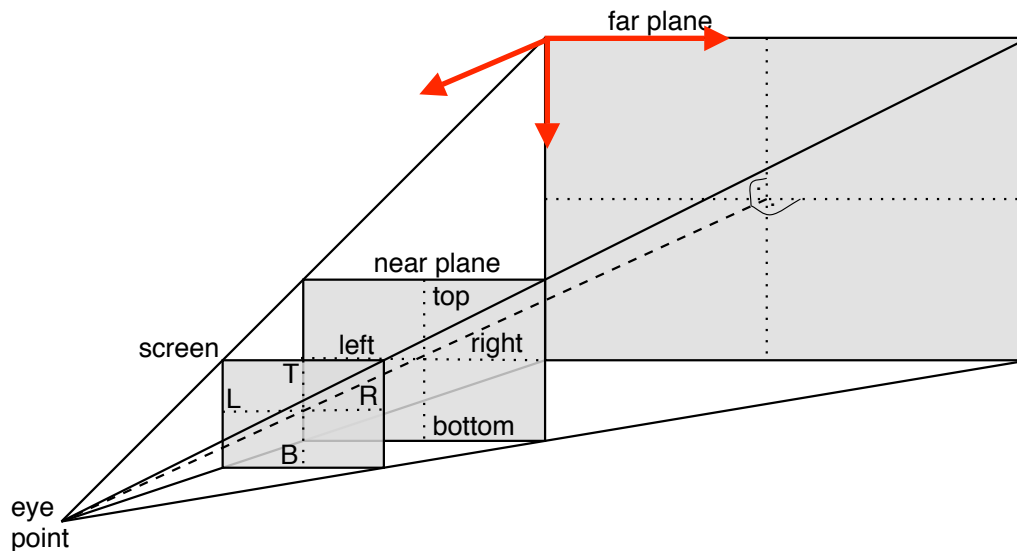
View vector is always perpendicular to the screen plane

Definitions - View vector



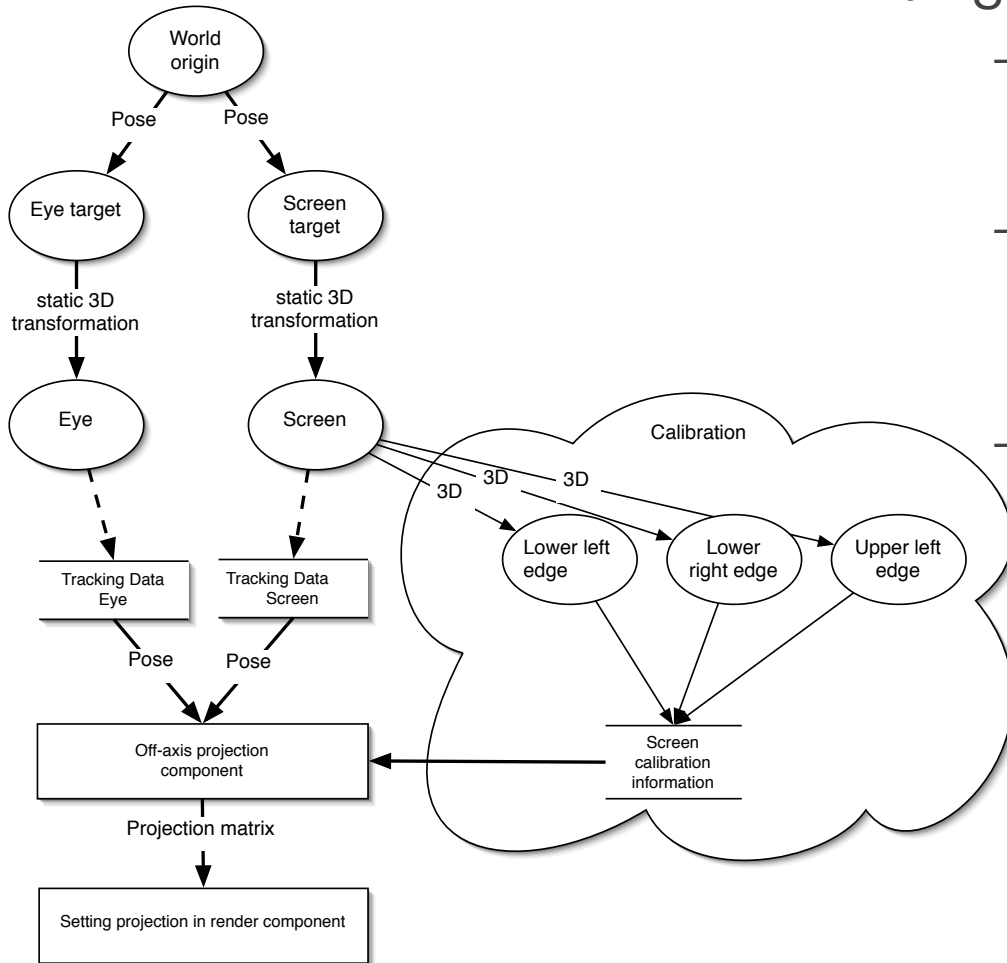
View vector is always perpendicular to the screen plane

Offaxis Projection



- construct transformation into coordinates local to the screen
 - view matrix
- distance from the view vector to the edges of the near plane are needed for perspective projection

About the implementation

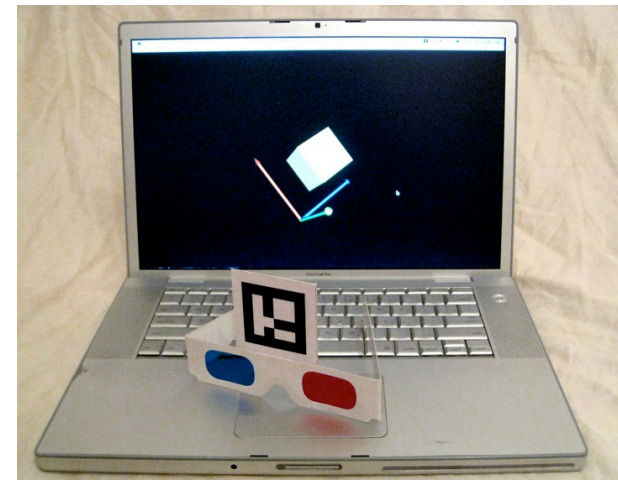
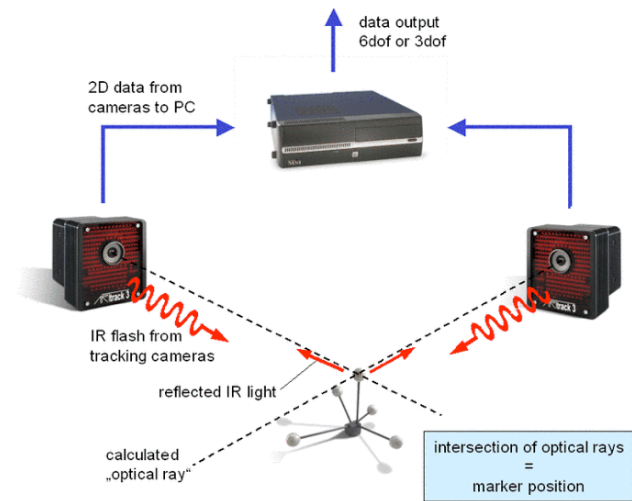


- Ubitrack component (OffaxisProjection)
 - Dynamic Inputs:
 - Eye point position
 - Screen position
 - Static Inputs:
 - Calibration results for screen corner offsets
 - Output:
 - Combined view and projection matrix for OpenGL

Development environments

- AR Lab
 - ART Tracking environment
 - portable TFT display
 - large static displays (beamer)

- “Homebrew CAVE environment”
 - Laptop with webcam and anaglyphic glasses with a paper marker



Encountered difficulties

- TCL Wrapper for Ubitrack
 - no support for callbacks from Ubitrack to TCL
- accurate screen calibration for large displays
 - command line utility for screen calibration
- eye calibration without SPAAM
- incorrect 2d projections hard to interpret/debug
 - stereoscopic mode with anaglyphic glasses
- TFT display too heavy
 - light wireless touchpad/portable see-through display

Encountered difficulties

- TCL Wrapper for Ubitrack
 - no support for callbacks from Ubitrack to TCL
- accurate screen calibration for large displays
 - command line utility for screen calibration
- eye calibration without SPAAM
- incorrect 2d projections hard to interpret/debug
 - stereoscopic mode with anaglyphic glasses
- TFT display too heavy
 - light wireless touchpad/portable see-through display
- addicting
 - never thought watching a virtual sheep could be fun ...

Byproducts

- TCL Wrapper for Ubitrack
- command line utility for screen calibration for the offaxis component configuration
- Coin3D based model loader/render component for Ubitrack
 - animation support
- component to inject projection matrixes in the Ubitrack OpenGL matrix stack
- anaglyphic rendering for Ubitrack

Future work

- better support for stereoscopic rendering in Ubitrack
- GUI for SPAAM eye calibration
- optimized versions for specific scenarios
- support for curved/resizable displays

Q&A