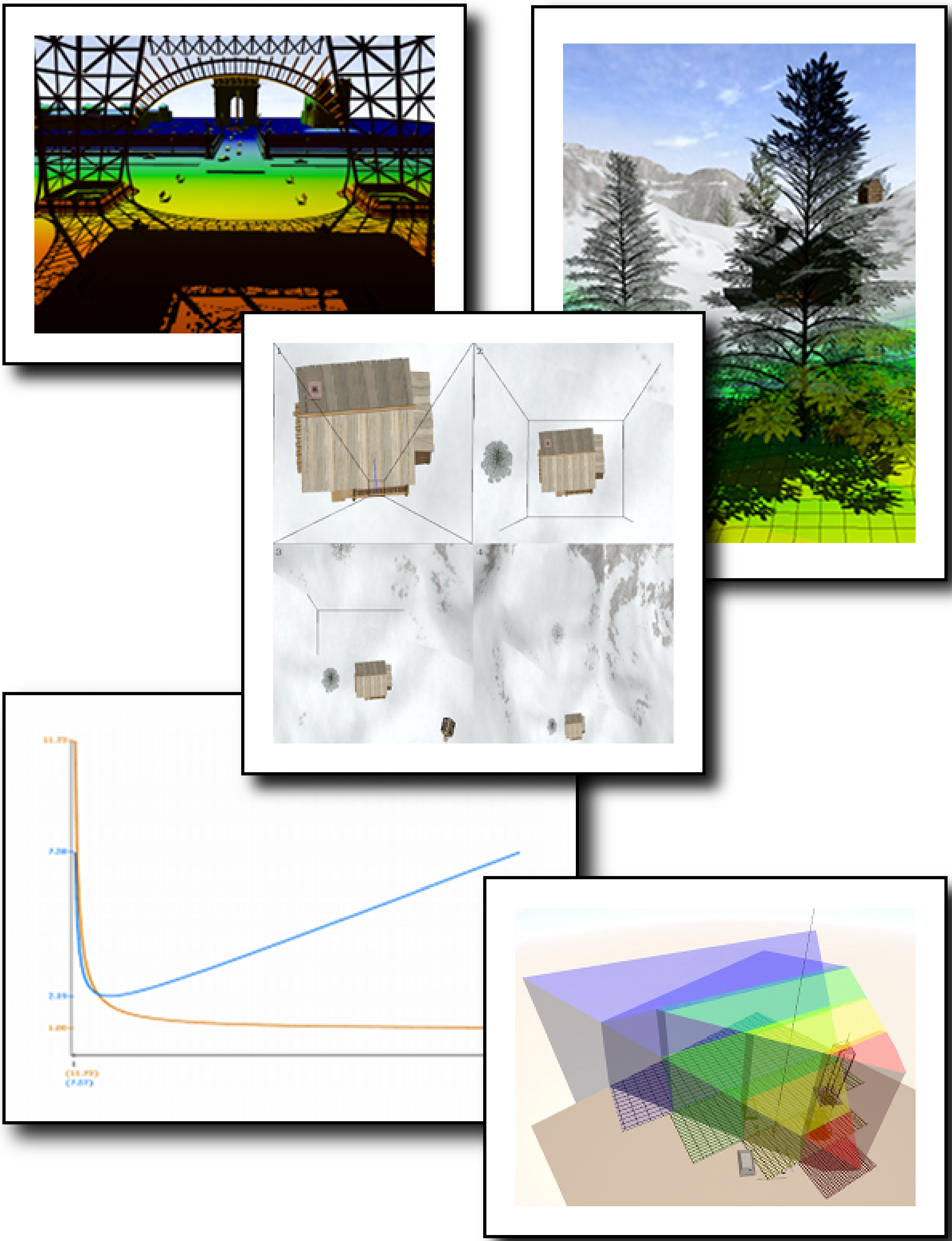
Shadow Mapping Framework

Robust Hard Shadows

USER GUIDE



Contents

CHAPTER 1

About this document

1.1 Introduction

This guide will explain the features and opportunities of the shadow mapping framework, which was developed within the work on Stingl's master thesis Robust hard shadows [Sti11]. It will help you to understand the displayed statistics and briefly describes the functions of the provided graphical user interface (GUI). An example screen shot of the introduced framework is shown in Figure 1.1.

1.2 Key features

The key features of the introduced shadow mapping framework are as follows:

- It implements several state of the art shadow mapping techniques.
- It provides interactive experimenting with different combinations and parameterizations of the implemented shadow mapping algorithms.
- It offers various interactive visualization tools to compare and analyze the results of your experiments.
- It features the opportunity to load and experiment with arbitrary scenarios.

The following section of this chapter will declare the structure of this guide and give a short outline of the contents of the following chapters.

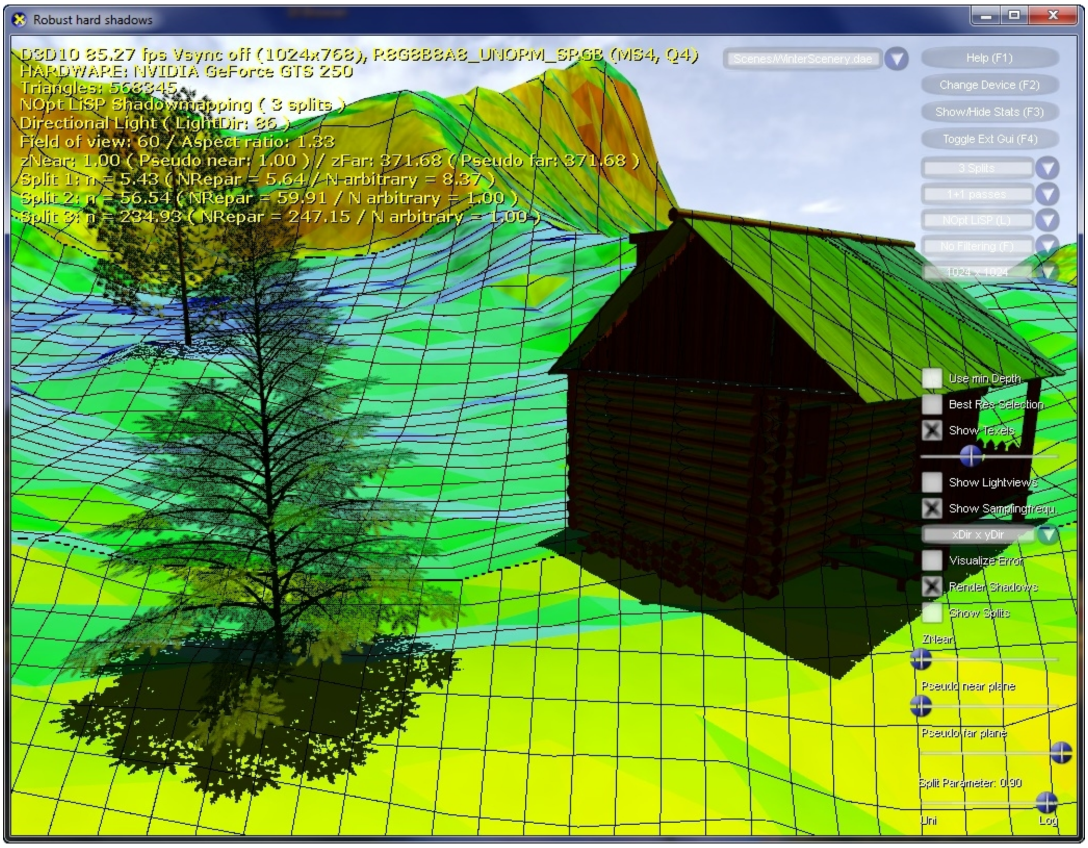


Figure 1.1: Screen shot of the introduced shadow mapping framework.

1.3 Structure of this guide

This guide is structured into different chapters as follows:

1. Chapter 2 at first lists the minimum requirements to run the framework. Then it describes the configuration opportunities and the possibilities to import custom scenes. At last it gives a short outline of the basic structure of the graphical user interface.

2. Chapter 3 offers a detailed description of the user interface elements and briefly explains the underlying functions. Furthermore, it lists all possible outputs that provide information about the current shadow mapping configuration of the framework.

CHAPTER 2

Configuration and structure

At first this chapter declares the minimum hardware and software requirements to run the application. In the following sections it explains the settings and values that can be adjusted before the application has been started and it describes the location of the configuration files and their names.

The GUI has been implemented by using the DirectX®Utility Library (DXUT), which comes with the DirectX®SDK [Mic11], and already provides many common GUI functionalities like for example *Buttons*, *Checkboxes*, *Radiobuttons* and much more. At the end of this chapter the main structure of the GUI and the functions of some basic GUI elements will be described.

2.1 Minimum requirements

To run the application, your PC/Laptop should at least meet the following requirements:

- Operating system: Microsoft®Windows®Vista or Microsoft®Windows®7
- Microsoft®DirectX®10 compatible graphics card
- 512MB video memory

2.2 Main configuration

The initialization file to change the main configuration of the application - `config.ini` - lies in the same directory as the execution binaries. It provides the opportunity to set the configuration baseline of the window resolution, the screen mode (windowed or fullscreen) and to configure the parameters for the viewer's perspective view frustum of the viewer. The possible configuration settings are shown in Table 2.1.

Window settings		
Setting	Enclosing tags	Type
Window width	`<width> </width>`	integer >0
Window height	`<height> </height>`	integer >0
Screen mode	`<fullscreen> </fullscreen>`	0: windowed 1: fullscreen

Camera view frustum		
Setting	Enclosing tags	Type
Field of view	`<fov> </fov>`	integer >0
Near plane distance	`<zNear> </zNear>`	integer >0
Far plane distance	`<zFar> </zFar>`	integer $>$ near plane distance

Table 2.1: Overview of possible configuration settings of the framework.

Note, that the GUI and the locations of its elements are optimized for a window resolution of *1024x768* pixels or higher. If you choose a lower resolution there might be some GUI elements outside of the displayed window and therefore not drawn.

2.3 Scene loading

As mentioned before, one feature of the introduced framework is the opportunity to load many arbitrary scenes at the program start without the need of recompiling the sources. In fact, the number of simultaneously loaded scenes is only limited by the available video memory. To specify the scenes that should be loaded at the program start, you just need to copy your model files into the directory *data/models* and adjust the corresponding initialization file. This initialization file is named `scenes.ini` and lies in the same directory as the execution binaries. It contains some additional commented out details of the compatible model format, which has been chosen to be Collada 1.4.1 [Khr11], and the list of filenames of the various scenes to be loaded.

If you want to experiment with your own scenarios and load them into the framework, it is very important to previously rotate respectively transform the entire scene corresponding to the coordinate system, shown in Figure 2.1.

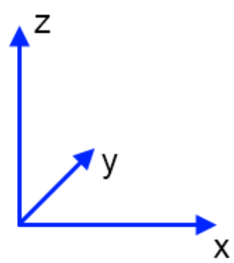


Figure 2.1: Coordinate system used for the scene import.

2.4 Main structure

Once the framework has been started, there are actually 5 main areas on the screen where the user can be provided with information about the current state of the application or can change its variables. An illustration of the interesting regions highlighted with different colors can be found in Figure 2.2. The statistics output (red highlighted area) can be de-/activated by mouse

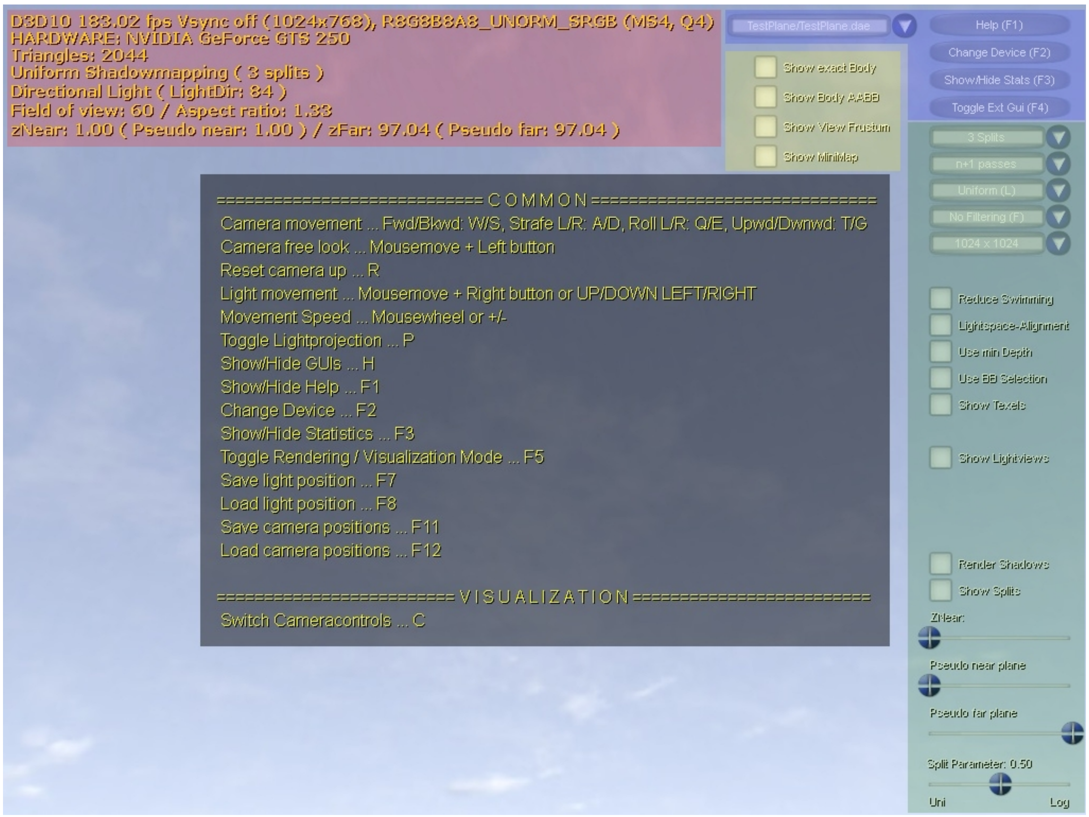


Figure 2.2: Illustration of the interesting information- and controlling areas in the introduced framework.

click on the button **Show/Hide Stats** or by the key **F3** and contains all relevant statistics and informations like for example frame rate, used graphics device and currently activated shadow mapping algorithms. The blue highlighted GUI area usually is always visible and offers the opportunity to

- switch the currently rendered/displayed scene.
- show an overlay with additional helping information on available keyboard and mouse inputs (black highlighted) - key **F1**.
- change the device/window settings - key **F2**.

- show the statistics like explained above (red highlighted area) - key **F3**.
- blend in an extended GUI to change the specific parameters of the current used shadow mapping techniques (green highlighted) - key **F4**.

The yellow highlighted GUI is only visible in the third person view, which and can be de-/activated by the key **F5**. A detailed description of this view will be given later in this guide.

CHAPTER 3

Features and outputs

At first this chapter shortly describes an overlay that provides some additional helping information. After that, the functions of the extended GUI will be explained in detail. This GUI offers almost all core functionalities of the framework and allows various experiments with the implemented shadow mapping techniques and features the different methods to analyze the current configuration.

The subsequent sections describe the outputs in the statistics area and the offered functions in the third person view resp. the visualization mode.

3.1 Help overlay

Figure 3.1 shows the essential keyboard and mouse controls. The controls below the headline *VISUALIZATION* are only in the third person view respectively the visualization mode available.

Another useful feature of the framework, as it can be seen in Figure 3.1, is the opportunity to save the light- respectively camera positions (keys **F7** and **F11**) and to reload them (keys **F8** and **F12**) at another point in time. The files, where the framework stores these entries, are located in the same directory as the executable and are named `c-positions.txt` and `l-positions.txt`.

3.2 Extended GUI functionalities

This section briefly describes the functions of the elements in the extended GUI. For further details about the actual implementation of the underlying functions it is referred to Stingl [Sti11].

It is important to know that some of the explained functionalities do not make sense respectively do not work for all configurations of the application, which is why the corresponding GUI

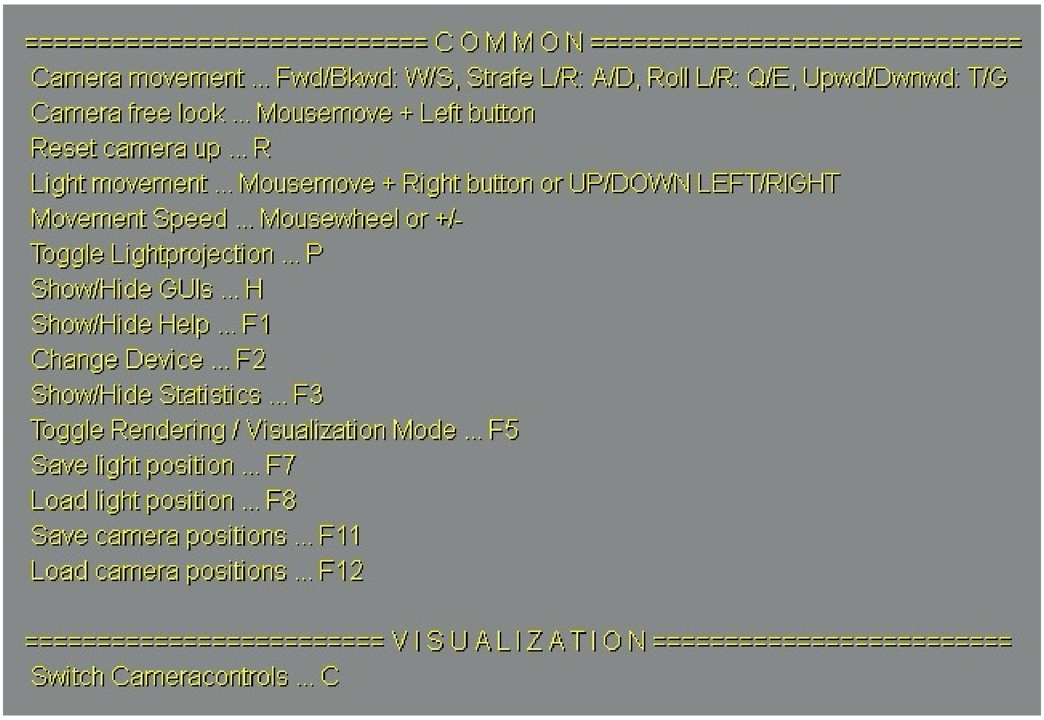


Figure 3.1: Help overlay showing the essential keyboard and mouse controls.

elements are hidden or disabled in such cases.

The drop-down menus, shown in Figure 3.2a, offer the following functionalities:

- Sets the number of z-partitioned splits between 1 and 16.
- In case of active z-partitioning, this menu allows choosing between the common rendering method, where each split shadow map is generated in one single pass, and *geometry cloning*, where all split shadow maps are generated in one pass by using the DirectX 10 geometry shader stage.
- Allows choosing between several different shadow map sample redistribution schemes. *Uniform* corresponds to standard shadow mapping without any redistribution of the shadow map samples. The methods *Nopt LiSP*, *NRepar LiSP* and *N arbitrary* can be summarized as *warping*-techniques.
- Allows choosing between various filtering techniques.
- Sets the shadow map resolution for each split.

Figure 3.2b shows the following features:

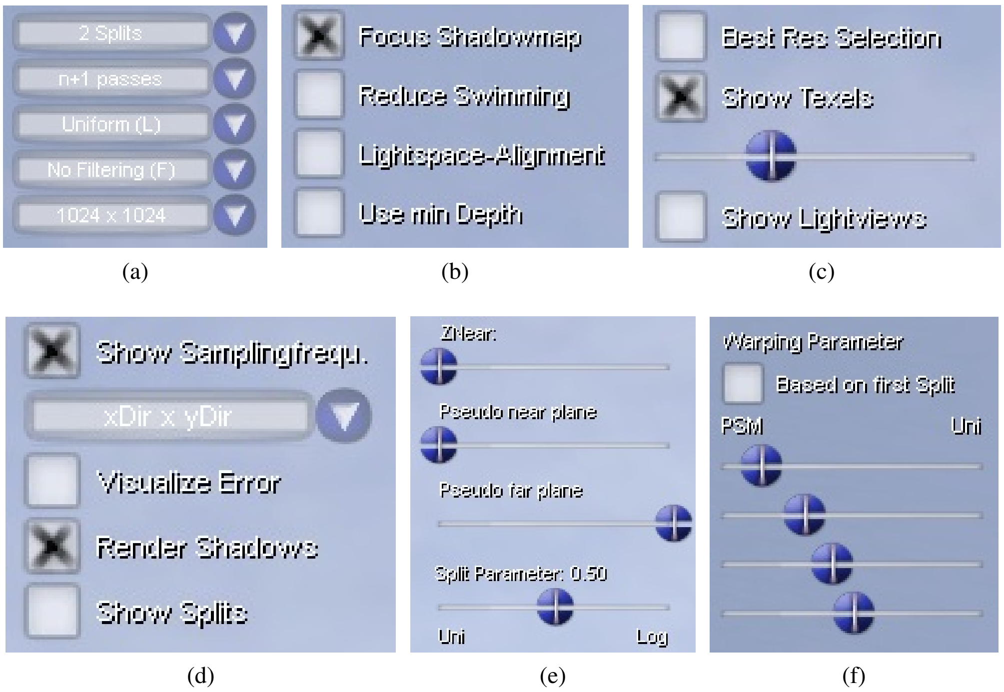


Figure 3.2: Overview of the GUI elements of the extended GUI.

- *Focus Shadowmap* - De-/Activates accurate focusing of the shadow map based on a computed intersection body of the current scene bounding box, the view frustum and the light frustum.
- *Reduce swimming* - De-/Activates a method to reduce the occurrence of ßhimmeringßhadow boundaries.
- *Lightspace-Alignment* - De-/Activates a rotation of the light-space (light's up vector) according to the current view direction.
- *Use min Depth* - De-/Activates a method to automatically adjust the near plane distance that will be used for the focusing of the shadow map.

The elements, shown Figure 3.2c, provide the following features:

- *Best Res Selection* - In case of z-partitioning, this check-box activates a split selection technique, which is based on the available shadow map resolution.
- *Show Texels* - De-/Activates the drawing of a grid that is based on the shadow map texels. The slider element below allows adjustments of the grid size.

- *Show Lightviews* - De-/Activates the drawing of a small map on the lower left that shows for each split the current light view including the (blue) view vector (only in the first split) and the current split frustum.

Figure 3.2d shows the following visualization features:

- *Show Samplingfrequ.* - De-/Activates the visualization of the sampling frequency via color overlay in the scene. See Figure 3.3 for the color mappings. The drop-down menu below allows choosing between the combination of both shadow map directions, or only the x- or y-direction of the shadow map.
- *Visualize Error* - De-/Activates the drawing of a small map on the lower left that shows the distribution of the perspective aliasing error in a 2-dimensional graph.
- *Render Shadows* - Dis-/Enables shadow rendering
- *Show Splits* - De-/Activates the visualization of the current set split scheme and the used split selection technique.

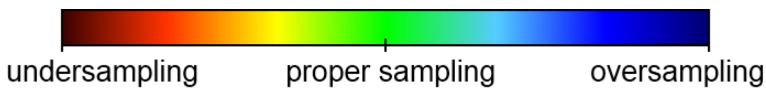


Figure 3.3: Color mapping for the visualization of the sampling frequency.

The slider elements, shown in Figure 3.2e, offer the opportunity to manually adjust the near plane distance (used for focusing), the pseudo near and pseudo far plane, and to continuously adapt the split scheme between uniform and logarithmic z-partitioning.

The GUI elements, shown in Figure 3.2f, are only visible if the chosen warping scheme corresponds to *N arbitrary*. They allow manual adaptation of the warping parameter for each split between perspective shadow mapping (warping parameter = near plane distance, $n = z_n$) and uniform shadow mapping ($n \to \infty$). In case of an automatically adjusted near plane distance, the slider element for the first split (on the top) will get disabled for user inputs, and automatically set according to the current near plane distance.

3.3 Statistics output

The output of the relevant statistics and information of the current configuration consists basically of 7 text lines (see Figure 3.4):

1. Output of the current frames per second. Output of the current window resolution. Output of the currently used frame-buffer format and the type of multi-sampling and its number of samples.
2. Vendor and model of the installed graphics card.

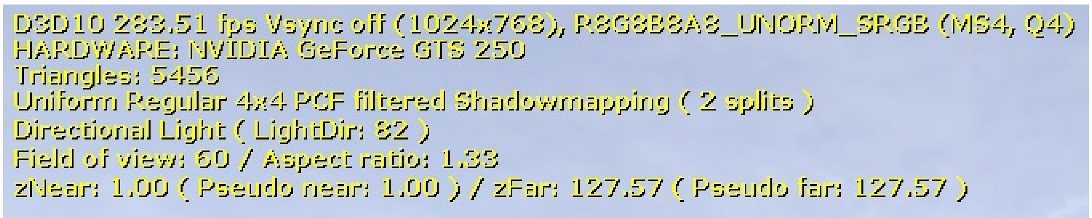


Figure 3.4: Output of all relevant statistics and information.

3. Amount of currently processed triangles, including all 3D rendering passes.
4. Details about the current used shadow mapping technique and, in case of applied z-partitioning, the number of splits. Further information about the different states of this line will be given later in this section.
5. Information about the current used type of light projection and the current angle between the view and light direction (*LightDir*).
6. Output of the current set field of view and the current aspect ratio (see also Section 2.2).
7. Output of the near plane distance and the far plane distance of the view frustum, currently used for the focusing.

In case of activated warping, a line is added for each split, printing the warping parameters for every available warping method simultaneously.

The 4^{th} line of the statistics output is assembled in the following way:

Type of shadow map redistribution - *Filtering method* - Shadowmapping - [(*Number of splits*)]

Type of shadow map redistribution can be

- *Uniform*: standard shadow mapping without any redistribution of the shadow map samples.
- *NOpt LiSP*: Light Space Perspective shadow mapping using the warping parameter n_{opt} (see Stingl [Sti11]).
- *Reparametrized LiSP*: Light Space Perspective shadow mapping using the warping parameter n_{repar} (see Stingl [Sti11]).
- *Arbitrarily warped*: warping based on the set slider value for the warping parameter.

Filtering method can be

- Empty: in case of deactivated filtering
- *Hardware PCF filtered*: 2x2 bilinear PCF
- *Regular YxY PCF filtered*: YxY PCF using a boxed filter kernel (Y stands for the number of samples)
- *Poisson Yx PCF filtered*: Yx PCF using samples on a Poisson Disc. (Y stands for the number of samples)

The brackets including the number of splits are only printed for activated z-partitioning.

3.4 Features of the visualization mode

The third person view or so called visualization mode sets the viewer position to another point of view. This view allows you to investigate the current shadow mapping configuration from an outside point of view. Furthermore, it is possible to display the intersection body, which is used for the focusing, the world-space axis aligned bounding box of this intersection body, the view frustum and a mini map on the lower left that shows the perspective view of the actual viewer's point of view. An example configuration can be seen in Figure 3.5.

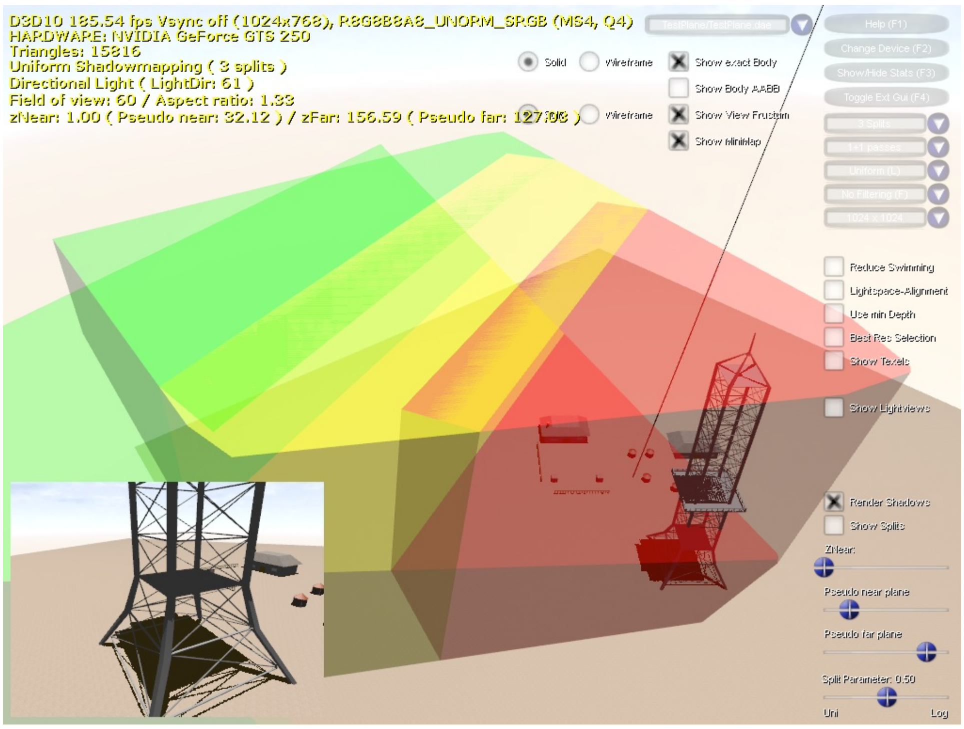


Figure 3.5: Example configuration of the visualization mode.

List of Figures

Bibliography

[Khr11] Khronos Group. COLLADA - Digital Asset Exchange Schema for Interactive 3D. http://www.khronos.org/collada/, 2011.

[Mic11] Microsoft Corporation. DirectX Developer Center. http://msdn.microsoft.com/en-us/directx/, 2011.

[Sti11] Martin Stingl. Robust hard shadows. Master's thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, September 2011. http://www.cg.tuwien.ac.at/research/publications/2011/Stingl_2011_RHS/.