

Volume and Surface Rendering of 3D Medical Datasets in Unity[®]

Soraia Figueiredo Paulo¹, Miguel Belo¹, Rafael Kuffner dos Anjos^{1,2,3}, Joaquim Armando Jorge^{1,2}, Daniel Simões Lopes^{1,2}

¹ INESC-ID, Lisboa, Portugal

² Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

³ FCSH, Universidade Nova de Lisboa

REngine consists of a complete Unity (C#) project for interactive volume rendering of 3D Textures and surface rendering of triangular meshes. Volume data is reconstructed from medical image datasets, such as Computed Tomography (CT) or Magnetic Resonance Imaging (MRI), and rendered using a raymarching shader. Surface renderings are created based on 3D meshes and traditional shading algorithms such as Blinn-Phong. This tutorial is a step-by-step demonstration of how to use this project to generate a 3D volume from 2D medical images and render 3D surfaces using Unity 3D game engine (<https://unity3d.com/>).

1. Download project

REngine is available for download at the following public repository <https://bitbucket.org/it-medex/volume-rendering-engine>.

On the left menu, click on *Downloads* (Figure 1) and then, *Download Repository*.

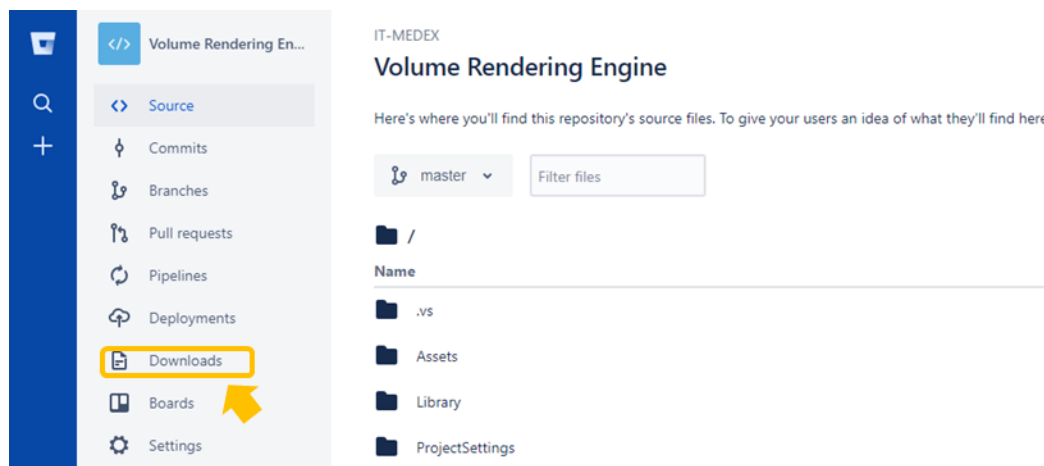


Figure 1. REngine Bitbucket Repository

2. Import into Unity

After downloading the .zip file, uncompress the project and open it using Unity. To do so, open the *Assets* folder and double-click on *scene* (Figure 2).

Name	Date modified	Type	Size
Resources	24-05-2018 14:15	File folder	
Loader.cs	24-05-2018 14:15	Visual C# Source f	8 KB
Loader.cs.meta	24-05-2018 14:15	META File	1 KB
Parser.cs	24-05-2018 14:15	Visual C# Source f	2 KB
Parser.cs.meta	24-05-2018 14:15	META File	1 KB
RayCaster	24-05-2018 14:15	SHADER File	5 KB
RayCaster.shader.meta	24-05-2018 14:15	META File	1 KB
Resources.meta	24-05-2018 14:15	META File	1 KB
scene	24-05-2018 14:15	Unity scene file	17 KB
scene.unity.meta	24-05-2018 14:15	META File	1 KB

Figure 2. REngine Unity Project – Assets folder

By default, this project will display the volume rendering of a CT dataset (once freely available at the OsiriX DICOM Image Library (<https://www.osirix-viewer.com/resources/dicom-image-library/>), alias name *PHENIX*) that uses the mapping function contained within the *TransferFunction.txt* file (Figure 3). This file consists of a list of Red, Green, Blue and Alpha (RGBA) colors that are assigned each voxel according to its original grayscale color intensity. All values are separated by commas. An example from one of the file's lines would be *1.0000, 0.8673, 0.6368, 0.7500, 180* which means that for all voxels with a grayscale color intensity of *180*, the new color will be *1.0000* (Red channel), *0.8673* (Green channel), *0.6368* (Blue channel), *0.7500* (Alpha channel). All RGBA values are on a scale from 0.0 to 1.0.

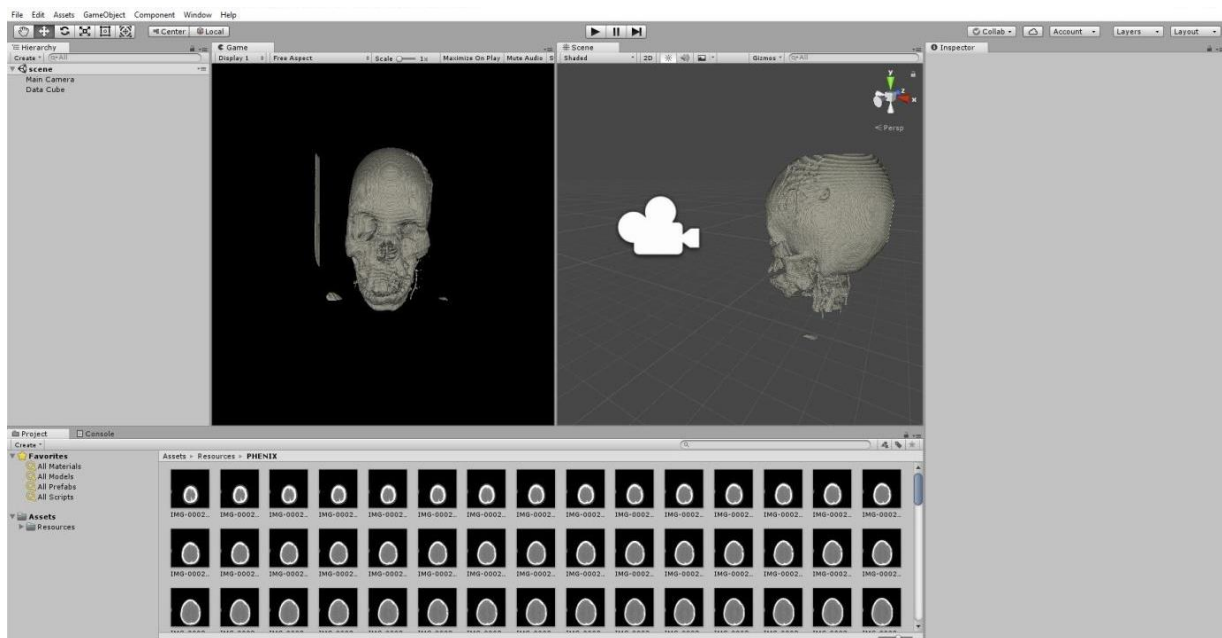


Figure 3. REngine Unity Project – Initial Display

3. Render your own image dataset

To illustrate the steps involved in generating a new volume rendering, we will use a different dataset from the OsiriX DICOM Image Library, alias name *INCISIX*, that is also available inside the REngine project.

3.1. Create a new folder

First, create a new folder with the name of your image dataset inside the project's *Resources*. Under the *Project* window, open the *Assets* folder and then open *Resources* (Figure 4).

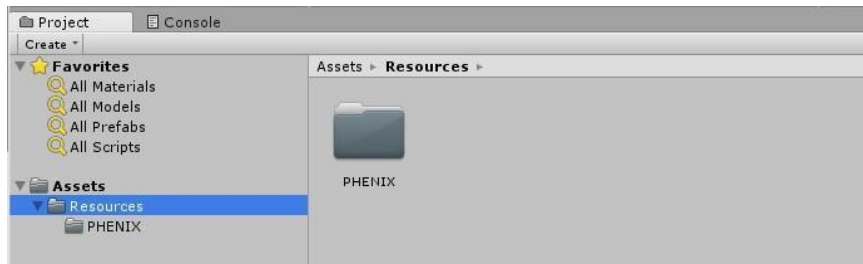


Figure 4. Project window – Resources folder

To create a new folder, click on the *Assets* menu, *Create* → *Folder* (Figure 5(a)). Rename the new folder (Figure 5(b)).

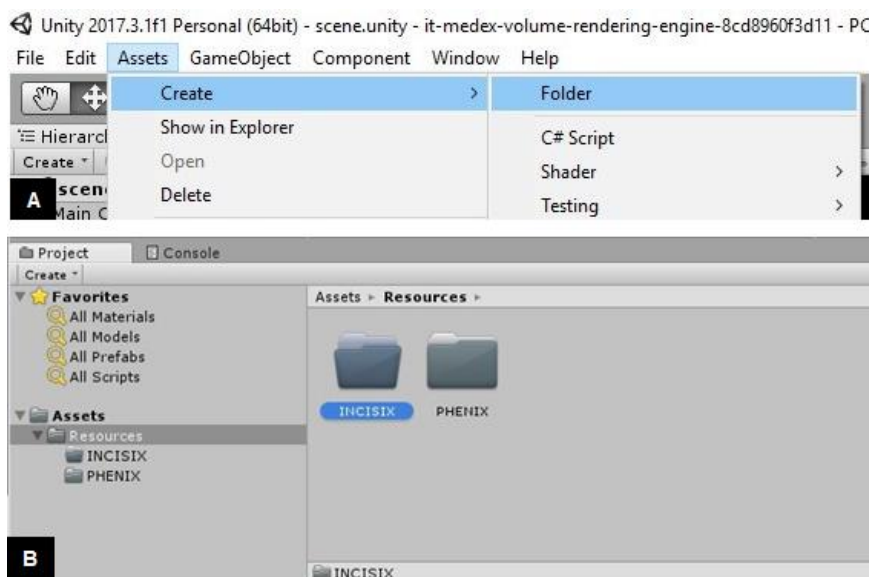


Figure 5. (a) How to create a new folder inside Resources; (b) New folder renamed - INCISIX

3.2. Import your image dataset

Drag and drop your 2D image dataset (*.jpg*, *.bmp* or *.png* format) into your new folder. Select all images and on the *Inspector* window activate the *Read/Write Enabled* option, and change the *Wrap Mode* from *Repeat* to *Clamp* (Figure 6). Press *Apply*.

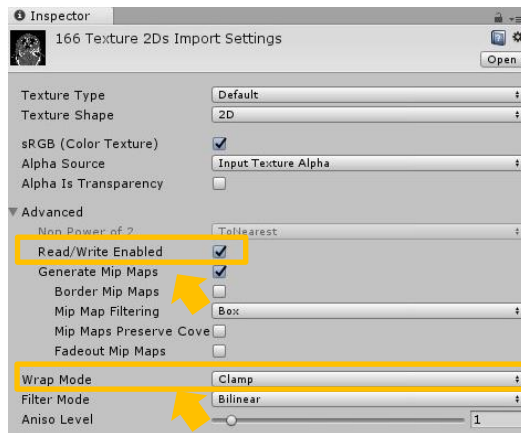


Figure 6. Inspector View – Import Settings for new 2D image dataset

3.3. Load your volume rendering

In the *Hierarchy* window, select the *Data Cube* gameObject. In the *Inspector* window, activate the *Loader.cs* script and fill in the data for your new image dataset (Figure 7). To create your new volume rendering provide the name of the folder from the project's Resources where your image dataset is (INCISIX), along with the filename of the 3D texture that is going to be generated (INCISIX_Texture) and save all changes. Press the play icon (upper bar, right at the middle of the window).

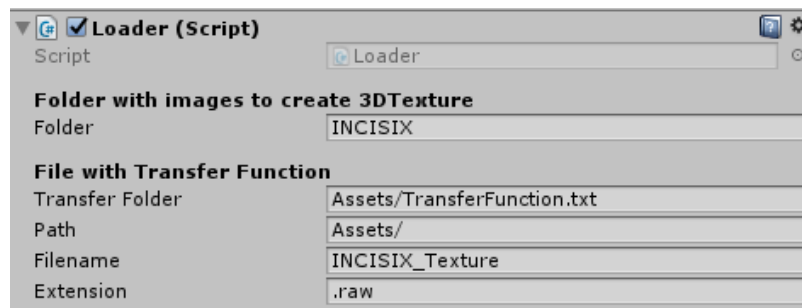


Figure 7. Loader script associated with the Data Cube gameObject

3.4. Visualize your volume rendering

On play mode, a new .raw file is created and displayed (Figure 8). The user can manipulate this volume by activating the *Transformer.cs* script associated to the *Data Cube*. Translation and rotation are applied by right or left-clicking, respectively, on the volume while dragging the mouse. Scale is controlled through the mouse scroll.

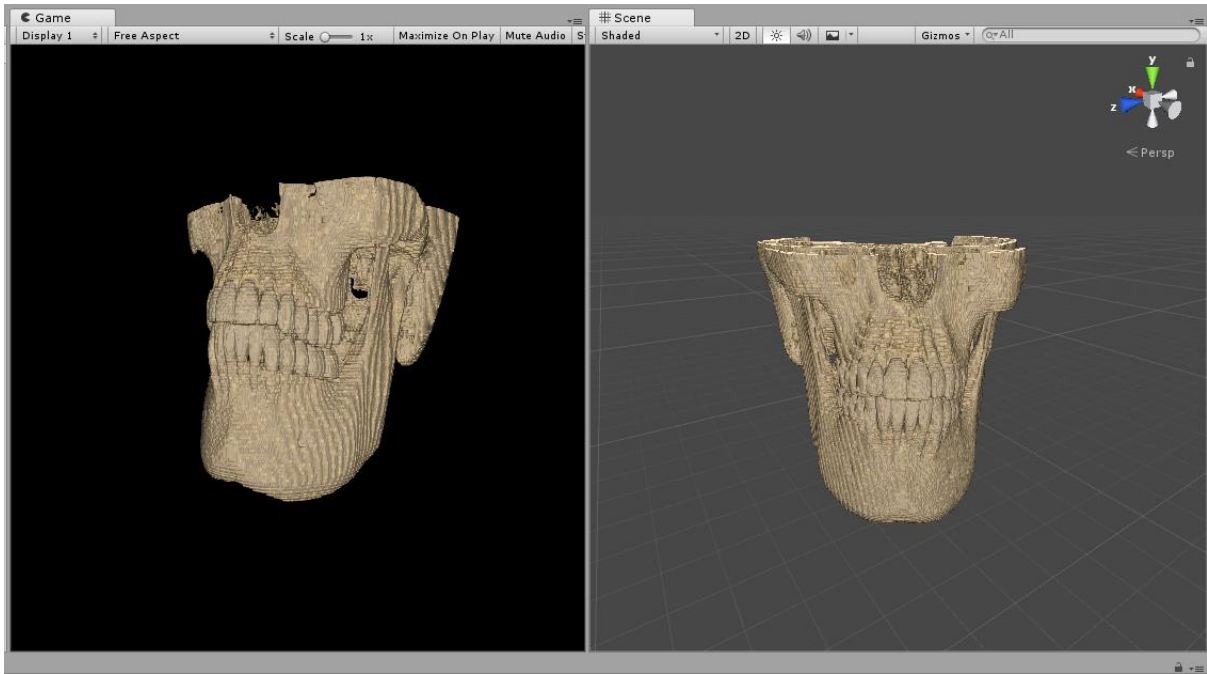


Figure 8. Game (left) and Scene (right) views - 3D Volume rendering of the INCISIX 2D image dataset

4. Render your own model

To demonstrate the process of creating a new surface rendering we will use a 3D dental implant model (available at Sketchup's 3D Warehouse <https://bit.ly/2vfT9bu>).

4.1. Import the model into Unity

On the *Assets* menu, click on *Import New Asset* to select the model you wish to import (Figure 9(a)). This model (*Implant+screw.skp*) will be displayed on the *Assets* window (Figure 9(b)).

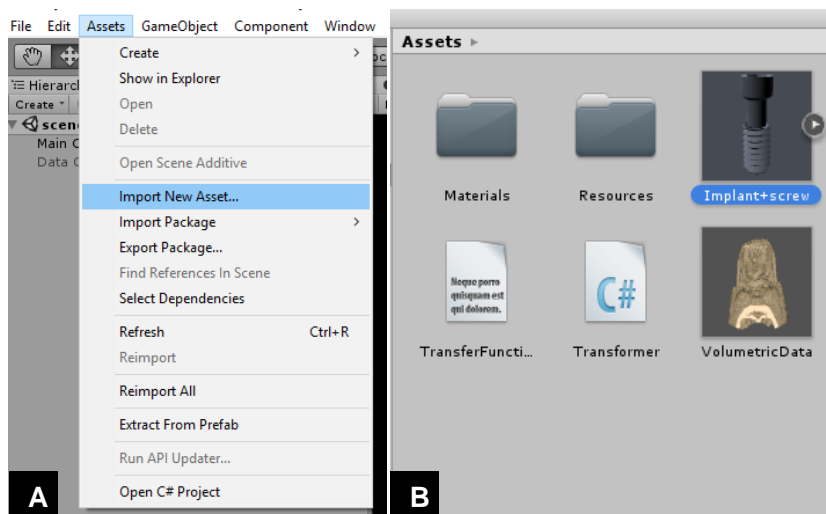


Figure 9. (a) How to import a new asset into the project; (b) New 3D model imported and displayed on the Assets window - *Implant+screw.skp*

To add the model to the scene, drag and drop it from the *Assets* window into the *Scene* window (Figure 10).

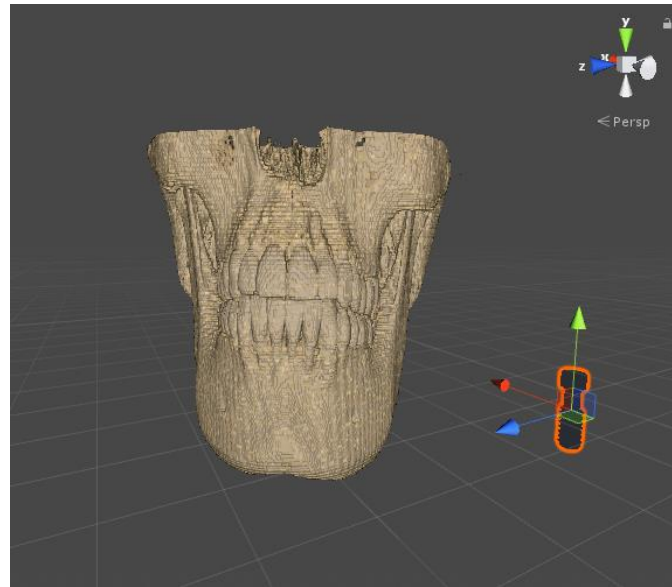


Figure 10. Add the 3D model to the scene - Scene View

4.2. Add a light component to the project

To enhance the visualization of your surface rendering, add a lighting feature to your project. To do so, right-click on the *Hierarchy* window, *Light*→ *Directional Light* (Figure 11).

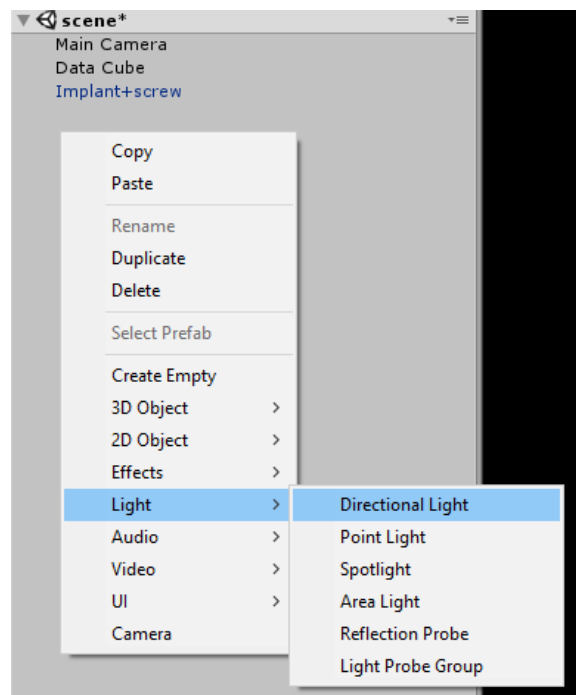


Figure 11. How to add a Directional Light to the scene - Hierarchy window

4.3. Visualize your surface and volume renderings

On play mode, visualize and manipulate the 3D surface and volume renderings (Figure 12).



Figure 12. 3D Surface and Volume Renderings: (a) 3D dental implant model; (b) and (c) INCISIX volume Rendering and 3D dental implant surface rendering - Game window