A. Manual for Simulation Usage

A.1. Installation

A.1.1. System Requirement

This simulation environment could be installed on different operating systems. What is mentioned below is the installation guide for Linux Ubuntu 16.04 LTS operating system.

A.1.2. Install Blender

Blender could be found under :
https://www.blender.org/download/
Installation instructions could be found under:
https://www.blender.org/manual/getting_started/installing/introduction.html#install-blender
Installed Version: 2.77

A.1.3. Installations to Import Open-Exr Files

The output of the radar simulation in Blender are images in **Open-EXR** format.
To import **Open-EXR** files to MATLAB follow the following instructions:

1. Install zlib from here:
   http://zlib.net/zlib-1.2.10.tar.gz
2. Install ilmbase from here:
   http://mirror2.klaus-uwe.me/nongnu//openexr/ilmbase-2.2.0.tar.gz
3. install openexr from here:
   http://mirror2.klaus-uwe.me/nongnu//openexr/openexr-2.2.0.tar.gz
4. Download the **open-exr-matlab-master** folder:
   https://codeload.github.com/skycaptain/openexr-matlab/zip/master
5. Run **make.m** in MATLAB

A.1.4. Enable the Add-on Export Animation Camera

To simulate camera movements, record camera positions and directions at different frames with key-frames.
The camera potion at each frame can be exported to a python file with the Add-on **Export**

Camera Animation.
To enable the add-on in Blender:

1. Open **User-Preferences** from the tab **File**
2. Select the tab **Add-ons** and the category **Import-Export**
3. Enable the Add-on by checking the box next to the add-on **Export-Import: Export Camera Animation** and Save the user settings

A.2. Blender Add-on Materials Toolbox

A.2.1. Install the Add-on Materials Toolbox

The code for the Add-on Materials Toolbox is located in the folder **Python_Scripts** and has the name **simulate_materials.py**.

To install the add-on in Blender:

1. Open **User-Preferences** from the tab **File**
2. Select the tab **Add-ons** and the category **Material**
3. Select **Install from File...** and navigate to the location of the python script
4. Select the python script file **simulate_materials.py**
5. The add on will be listed in the list of the add-on for Material category
6. Enable the Add-on by checking the box next to it and Save the user settings

The Add-on can be disabled and removed using the same steps.

A.2.2. Usage of Materials Toolbox

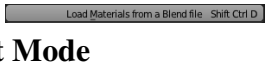
To use the Materials Toolbox:

1. Select **Object Mode** 

2. Press *CTRL+SHIFT+M*
3. The Add-on Menu will appear as in figure A.1

Load Predefined Materials

1. To load the predefines materials list select 

in the Materials Toolbox Menu or press *CTRL+SHIFT+L* in **Object Mode**
2. The predefined Materials will be added to the materials list as in figure A.2
3. To load predefined materials from another blend file select 

in the Materials Toolbox Menu press *CTRL+SHIFT+D* in **Object Mode**
4. Select the .blend file to add materials from

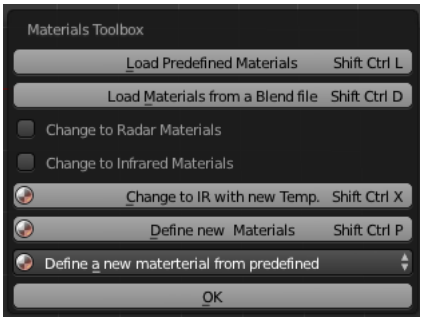


Figure A.1.: Materials Toolbox Menu

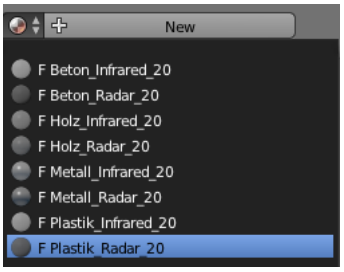


Figure A.2.: The loaded Materials list

Change to Radar or Infrared

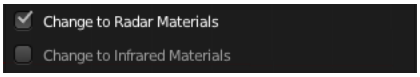


Figure A.3.: Change to Radar or Infrared Scene Settings

1. Check Change to Radar to switch the materials assigned to all objects in the scene to radar-specific materials
2. The light source (Spot lamp) used for radar simulation **Radar_Antenna** will also be visible and render-able
3. Check Change to Infrared to switch the materials assigned to all objects in the scene to infrared-specific materials
4. The light source (Area lamp) used as external light source for infrared simulation **Infrared_Lamp** will also be visible and render-able

Notes :

1. Use the default coordinate system of Blender, with x,y-coordinates the location in the image and z the depth information (distance object-camera).
2. The materials assigned to the objects has to have the same name-format e.g. **Metal_Radar_20**, indicating the name of the sensor (Radar/Infrared) and the temperature of the object (default: 20°).
3. To switch from one sensor to the other one the materials has to have the defined name-format and the material exists for both sensors (Radar and Infrared) for the desired temperature in the materials list.

4. The light source for radar has to be defined in the scene with the name **Radar_Antenna**, the add-on then sets the right size, location and rotation to fit the camera path.
5. The light source for infrared has to be defined in the scene with the name **Infrared_Lamp** and the desired location and direction.

Change to Infrared with a new temperature

1. To set a new temperature for infrared materials select: Change to IR with new Temp. Shift Ctrl X in the Materials Toolbox Menu or press *CTRL+SHIFT+X* in **Object Mode**
2. Choose the desired temperature in the Menu in figure A.4

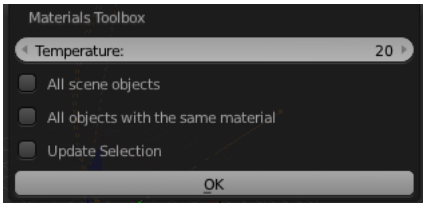


Figure A.4.: Change to Infrared with a new Temperature

3. Check **All scene objects** to change the temperature for all objects in the scene
4. Check **All object with the same Material** to change the temperature for all objects, which have the same material assigned to them as the selected object
5. Check **Update Selection** to change the temperature for the selected object only

Define new Materials

1. To define new materials select: Define new Materials Shift Ctrl P in the Materials Toolbox Menu or press *CTRL+SHIFT+P* in **Object Mode**
2. Define the materials name and the set the materials parameter for radar and infrared in the new materials' menu in figure A.5

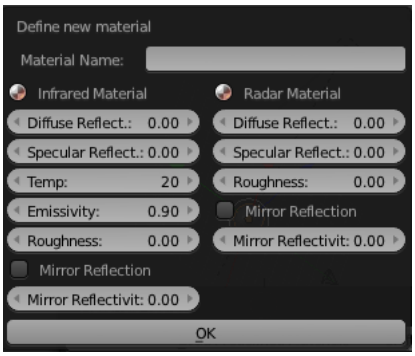


Figure A.5.: Define new Materials Menu

3. Two new materials one for each sensor will be added to the list
4. To define new materials by reusing parameters from other predefined materials select: in the Materials Toolbox Menu
5. Select a material from the list to reuse its parameters
6. The new materials' menu in figure A.5 will appear filled with the name and the parameters of the selected material
7. Change the desired parameters and define a new name for the material
8. Two new materials one for each sensor will be added to the list

Notes :

1. The newly defined materials will have the specific name format needed, so that they can be used for other functionalities of the Materials Toolbox, e.g. if the entered material name is **newname** and the defined temperature is **25**, the newly added materials will have the names **newname_Radar_25** and **newname_Infrared_25**
2. A new name or a new temperature has to be set for each newly defined material. If an already used material's name and temperature are chosen, the newly defined materials will add *.001* to the preexisting material's name. This name format cannot be used for other functionalities in the Materials Toolbox.
3. Save the .blend file To expand the materials list in the current file with the newly defined materials.

A.3. Usage of the Synchronized Simulation Chain

A.3.1. Run the Synchronized Simulation in MATLAB

To run the synchronized simulation in MATLAB for a defined Blender-Scene, run the m-file **simulation_main** after setting the following user-defined inputs: In the m-script **defineDirectories.m** define the following directories:

1. Blend File:
   Enter the path for the .blend file with the desired scene for simulation for the variable **directories.blendfilepath**
2. Python Files:
   The python scripts responsible for generating the sensor outputs from Blender in the needed formats are located in the folder **Python_Scripts**.
   Enter the path of the scripts **simulate_radar.py** and **simulate_infrared.py** for the variables **directories.radarpythonfile** and **directories.irpythonfile** to run the scripts from MATLAB.
3. Output Files:
   Enter the directories of the output files, in which Blender should save the rendered output images of each sensor in the variables **directories.outputfile_radar** and **directories.outputfile_ir**.
   Enter the directory, where the Radar Renders shall be saved, when converted to

.mat files in the variable **directories.radarmatfiles**. Default: **Radar_Renders** , **Infrared_Renders** and **Exr_to_mat_files**

4. Filenames:
   Define the names for the rendered images in the variables **directories.filename_radar** and **directories.filename_ir**. The radar simulation provide two images for the first frame with the name **filename_radar_z0001.exr** and **filename_radar_out0001.exr** . The infrared simulation rendered one image per frame with the name **filename_ir.hdr**
5. Blender Path:
   To run the Blender application from MATLAB enter the directory, where Blender is installed in the variable **directories.Blender_path**. If Blender is installed in the default path, enter only **blender** for this variable.
6. Results Path:
   Enter the path of the folder, where the simulation results should be saved in the variable **directories.results**, Default: The folder **simulationResults**.

In the m-script **defineSimulationSettings.m** define the following scene settings:

1. Radar Images Resolution:
   The desired images resolution of the rendered radar images can defined. Enter the number of horizontal pixels in the variable **camera.radar_numPx_x** and the number of vertical pixels in the variable **camera.radar_numPx_y**
2. Infrared Images Resolution:
   The desired images resolution of the rendered infrared images can defined. Enter the number of horizontal pixels in the variable **Camera.ir_numPx_x** and the number of vertical pixels in the variable **camera.ir_numPx_y**
3. Number of Frames:
   Enter the number of frames in the simulation for the variable **camera.numofFrames**
4. Frames per Second:
   Define the desired frame rate per second in the variable **camera.fps**, to calculate the speed of the camera movement.
5. Scaling Factor:
   Set the scaling factor to convert the Blender units in *m* in the variable **camera.scale**. The distances will be calculated as follows: 1BU = $\frac{1}{camera.scale}$
6. Camera data:
   Enter the camera focal length in the variable **camera.focalLen** and the horizontal size of the camera sensor area in mm in the variable **camera.dimension_width**. These are used to calculate the camera's field of view and the pixel size.

Enter the following radar modulation settings in the m-script **defineRadar.m**"

1. Antenna Parameters:
   Enter the Antenna Gain in dB in the variable **radarParameters.antenna_gain**.
   Define the antenna azimuth and elevation beam width in the variables:
   radarParameters.antenna_three_dB_x
   radarParameters.antenna_three_dB_y.

2. Modulation Parameters:
   The predefined radar modulation settings can be modified by the user by setting the values for the following radar parameters:

The center frequency	**radarParameters.f_c**
The Bandwidth	**radarParameters.B**
The chirp duration	**radarParameters.T**
The sampling frequency	**radarParameters.f_s**
The pulse repetition interval	**radarParameters.Tp**
The total number of pulses	**radarParameters.Np**
The number of samples per pulse	**radarParameters.Ns**
The antennas position matrix in terms of the wavelength	**radarParameters.P**

Notes :
The defined values in MATLAB overwrite any other definitions in Blender, eg. the Resolution of the rendered images or the Camera data.

The simulation runs in Blender automatically and continues in MATLAB to generate the following outputs:

1. For each frame an infrared image with the name **filename_ir.hdr** is rendered. All images are imported in MATLAB and edited to a video after a light brightness correction. An infrared video with the name **filename_ir.avi** is the output of the infrared simulation. The output video is located in the folder **simulationResults**.
2. For each frame in the simulation a .mat file is generated with the name **filename_radar*n*_out.mat**, with *n* indicating the frame number.
3. Each .mat-file contains list of the radar detections with five columns containing the following information:
   a) The ranges list in the 1.column, with an estimated range value for each target
   b) The velocities list in the 2.column, with an estimated velocity value for each target
   c) The amplitude list in the 3.column, with an estimated velocity value for each target
   d) The azimuth angels list in the 4.column, with an estimated azimuth value for each target
   e) The elevation angels list in the 5.column, with an estimated elevation value for each target

A.4. Troubleshooting Guide

This section presents a troubleshooting guide for some issues, that could occur while using and installing the simulation environment.

A.4.1. Scene Modeling Issues

In case of errors considering the design of the scene and the materials definitions, the following solutions could be tried:

1. Check the coordinate system of the designed scene. Use the default coordinate system of Blender, with x,y-coordinates the location in the image and z the depth information (distance object-camera).

2. Check the name-format of the assigned materials as described in sec. A.2.2. To switch from one sensor to the other one the materials has to have the defined name-format and the material exists for both sensors (Radar and Infrared) for the desired temperature in the materials list.

3. Check, that the name of the light source (Spot lamp) used for the radar scene is **Radar_Antenna**. The light source for the infrared scene (Area Lamp) should be names **Infrared_Lamp**.

4. Use the Materials Toolbox to add new material: The newly defined materials will have the specific name format needed, so that they can be used for other functionalities of the Materials Toolbox, e.g. if the entered material name is **newname** and the defined temperature is **25**, the newly added materials will have the names **newname_Radar_25** and **newname_Infrared_25**

5. A new name or a new temperature has to be set for each newly defined material. If an already used material's name and temperature are chosen, the newly defined materials will add *.001* to the preexisting material's name. This name format cannot be used for other functionalities in the Materials Toolbox.

6. Use The Materials Toolbox to switch from the radar scene to infrared scene and back at least once to check that the materials and light sources are assigned correctly.

7. Check the chosen the scaling factor to convert the Blender units in m in the variable **camera.scale**. The distances will be calculated as follows: 1BU = $\frac{1}{camera.scale}$

A.4.2. Blender Renders Issues

In case of errors considering the Blender rendering, the following solution suggestions could be tried:

1. Check the entered pass to Blender and the .blend scene in the script **defineDirectories.m** for the variable **directories.blendfilepath**

2. Check the entered path of the folder **Python_Scripts**. The path of the scripts **simulate_radar.py** and **simulate_infrared.py** for the variables **directories.radarpythonfile** and **directories.irpythonfile** to run the scripts from MATLAB.

3. Check the Blender version installed and install the latest version (Version tested: v2.77)

4. Check the entered Blender Path for the variable **directories.Blender_path** in the m-script **defineDirectories.m**. If Blender is installed in the default path, enter only **blender** for this variable.

5. Check the installation of the Materials Toolbox add-on described in section A.2.1 and the installation of the add-on Export Animation Camera in section A.1.4.