Reel or Real

Computer Graphics, Autumn 2021

Specification for Assignment 1

Alexandros Keros, Kartic Subr, George Kamaras

Due: 12/10/2021 (5pm)

(a) Photograph (b) Composited image

Figure 1: Augmented reality by rendering a virtual object into a real scene.

Edinburgh-based Reel or Real Studios is looking to recruit interns to help with their next VFX blockbuster. They are specifically looking for students who can operate independently "hackathon style" to produce prototypes. Since Computer Graphics pipelines consist of a number of tools, they have invited interested applicants to solve the following task: Starting with a photograph of a real scene, possibly taken with a mobile phone camera, augment it with computer graphics imagery created using a photorealistic renderer. Since the company logo is a pen, you are asked to include a real pen in the photograph (not shown in the above example). The main requirement is that the rendered image exhibits lighting, reflections and shadows that are consistent with the viewer's expectation of the real scene (*physical accuracy*). The final output is a single photorealistic image that contains a pen and a 3D computer generated (*rendered*) object(s) overlaid convincingly (or *composited*) on a photograph.

Workflow

The suggested workflow for this task is:

1. Organise an interesting scene.
2. Record the relative positions of interesting objects and the camera.
3. Take a photograph of the scene from the recorded camera position.
4. take pictures of the interesting textures.
5. Roughly model the interesting objects (including textures) using a 3D modelling tool.
6. Place a virtual camera in the scene (using the recorded position).
7. Choose a virtual model (3D mesh) that you would like to insert in the scene.
8. Place the mesh in the modelled scene using the tool.
9. Export the scene in a format that can be rendered using PBRT (version 3).
10. Check the exported file to see that light sources and the camera are included.
11. Render the scene using PBRT (version 3 is recommended).
12. Composite the rendered image of the mesh into the photograph taken in step 3.

Tools

You are required to use the *Physically Based Renderer PBRT* to generate the synthetic objects/scene. In addition, you may use the following tools/resources:

- *Artec3D* online .ply library: `https://www.artec3d.com/3d-models/ply`
- *Blender 3D* creation suite: `https://www.blender.org/`
- *Blender exporter for PBRT* Blender add-on:
  `https://github.com/stig-atle/io_scene_pbrt`
- *Luminance HDR* tonemapping software: `http://qtpfsgui.sourceforge.net/`
- *Gimp* image manipulation program: `https://www.gimp.org/`

What To Submit

Please submit a report illustrating and describing each of the steps in the workflow. Include a qualitative assessment of your result at each step. A submission of only the final output without explanations or intermediate steps will only receive half the credit. The report should be in `.pdf` format. All figures should be numbered, annotated, referenced and clearly visible.

In addition to the report, your submitted `.zip` file should contain the following images:

1. *A composited image.* Ensure that shadows, reflections, objects occlusions and colours are consistent and natural.

2. *Intermediate images,* including:

   - The initial, *real-world photograph*
   - *Photos of textures* (perceived surfaces of real-world objects) that are necessary to produce a believable result. Textures might appear on reflections, in the background of synthetic objects that cast shadows on them, or might influence shadows and ways light interacts with surfaces of synthetic objects.
   - If a 3D modelling program is used, such as Blender, *screenshots* illustrating the modelled scene that will later produce the synthetic objects.
   - *Renderings,* as generated by *PBRT*, as well as *color-corrected* (tonemapped) versions.

3. The *PBRT input file(s)* used to render the synthetic image.

All necessary files, appropriately named, should be compressed into a `.zip` file and submitted via the *Learn* platform (`https://learn.ed.ac.uk`) as the first assignment for the course.

Marking Scheme

A total of 100 points are assigned for this assignment but your marks will be halved, since this assignment's contribution to your grade for the course will be 50%. The marking scheme is described below. Numbers in parentheses indicate points assigned for each specific task. Each student is expected to design a unique scene and object(s). **Do not copy the example shown in this document or from other submissions**. You should provide evidence of each of the following points in your report, since marking will be based on the report. Your marks will depend on the degree of difficulty in the scene. e.g. if the object is a cube and its shadow is a square (which you could have just as well hand-drawn) you are unlikely to receive the full marks for the 'shadows' section. Remember, the objective is not to game this specification but to 'Wow!' the recruiters at Reel or Real.

- *PBRT*
  - Installation (5)
  - Generating a test image (5)
- Approximate *model* of a scene, necessary to produce photorealistic synthetic objects
  - Recreating real-world lighting as in the original photograph (5)
  - "Proxy" model(s) of real-world object(s) that affect rendered objects (10)
  - Textures on real objects which appear on rendered object(s) (20)
- *Rendering* synthetic object
  - Choose a synthetic object model from a 3D model library (`.ply` files) (5)
  - Render a synthetic scene with the above parameters (5)

- *Compositing / Merging* of synthetic objects on the real-world image
  - Photometric consistency, i.e. light intensity, direction, colour, shadows (10)
  - Seamless positional compositing, e.g. no mismatched edges (5)
  - Photorealistic shadows of rendered object on real scene (5)
  - Photorealistic shadows of real objects on rendered elements (5)
  - Reflections of textured objects on rendered objects (reflective material) (5)
  - Multiple rendered objects (5)
- Creative customisation. Add one other novel effect to your render. e.g. reflections between rendered objects, translucency, smoke/fog, etc. (10)

Example

In this section, we outline the generation process of the example composited image of Figure 1b. Our goal with that, is to provide a sample workflow, with tips and tricks, for approaching such problems.

1. *Planning* the scene
   - (a) We captured a real-world image (Figure 1a) exhibiting interesting visual properties, such as rough, difficult to render, textures (ground plane, box and book designs), complex objects (glass jar filled with coins) and distinct hard shadows cast from real-world objects. We planned to position the rendered objects in the middle of the image, to maximise their interactions with real-world elements. We made sure that interesting visual properties will be present in the synthetic scene, such as shadows cast from real-world objects onto the rendered ones, shadows of rendered objects remaining clearly visible, and reflections of real-world objects appearing onto the rendered elements.
   - (b) We identified the textures that will influence the appearance of the rendered objects. Here, the surfaces of the book and the box are separately photographed so that they can be used as textures on artificial proxy objects while modelling the synthetic scene.
   - (c) We selected the rendered objects, and their materials, to be used for the new scene, making sure that they exhibit interesting visual properties, such as complex shadows and reflections. We decided to place a mirror icosahedron underneath an antelope skull, which is retrieved from an online 3D model library (`https://www.artec3d.com/3d-models/damaliscus-korrigum`).
2. *Modelling* the scene. Two approaches exist for modelling the desired scene:
   - One approach is to directly use *PBRT* and, by using built-in shapes and triangular meshes, define the scene to be rendered.
   - An *alternative* approach is to use a 3D modelling software that interfaces with PBRT (see *Tools To Use* section). *Blender*, which is freely available, alongside its respective *PBRT exporter* (see *Tools To Use* section) were used for this assignment's example scene.

First, we modelled the scene in Blender (Figure 2), making sure that materials are appropriately defined, and desirable textures appropriately mapped. Then, we exported the appropriate *PBRT* input file.

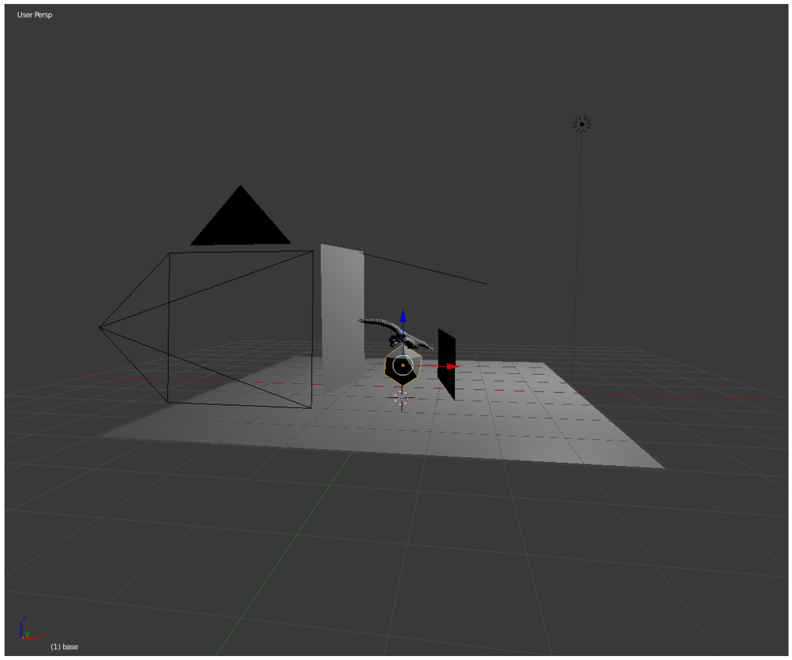


Figure 2: Blender model of the example scene.

3. *Rendering* the scene

   We used the files produced by *Blender-to-PBRT exporter* to render the synthetic scene. Note that the exporter does not define a light source, so the *PBRT* input file should be appropriately edited in order to get the desirable results (Figure 3).

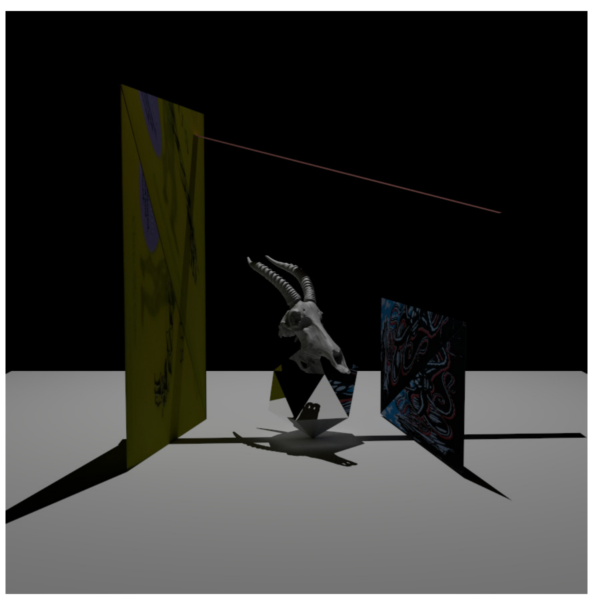


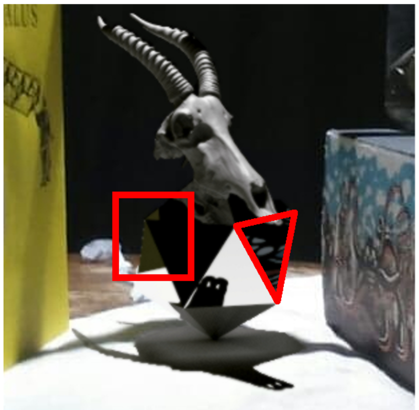
Figure 3: Example rendered scene.

4. *Compositing* the scene
    (a) We adjusted the colour histogram and intensity of the produced High Dynamic Range (HDR) rendering (*tonemapping*) with the freely available *Luminance HDR* software (see *Tools To Use* section) to match the real-world scene as closely as possible.
    (b) We used *Gimp* image editing software (see *Tools To Use* section) to composite the virtual objects with the real-world photograph. We made sure to further adjust colour values, where necessary, and that objects with shadows produce a photorealistic result.

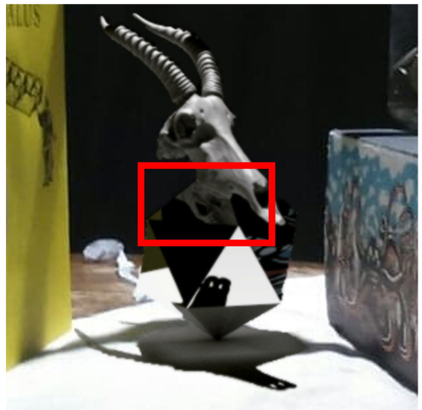
Figure 4: Zoomed area of final composited image (Figure 1b).

Assuming the above workflow, let us zoom-in at an area of the final composited image (Figure 4), in order to take a closer look in some of the interesting visual features that we can get from the real-world and synthetic objects interaction.

- Reflections of real objects are visible on the reflective surfaces of the rendered objects (Figure 5a). This is achieved by modelling the necessary real-world objects using the appropriate shapes (here, we used planes) and attaching textures (book cover, box design).
- Synthetic objects coexist in a photorealistic way, visible from the reflection of the underside of the skull on the face of the mirrored icosahedron (Figure 5b).
- Shadows of synthetic objects are reproduced and blend realistically in the real-world scene (Figure 5c) due to modelling the overhanging ribbon. Since they have been rendered in a photorealistic way, a complex instance of an idol of the shadow of the skull is visible on a face of the icosahedron (Figure 5d).
- Shadows of real-world objects are seen to be cast onto the synthetic objects, such as the horns of the skull (Figure 5e).



(a) Real objects' reflections



(b) Virtual object's reflection

(c) Virtual objects' shadows

(d) Reflection of virtual object's shadow

(e) Shadows of real objects on virtual objects

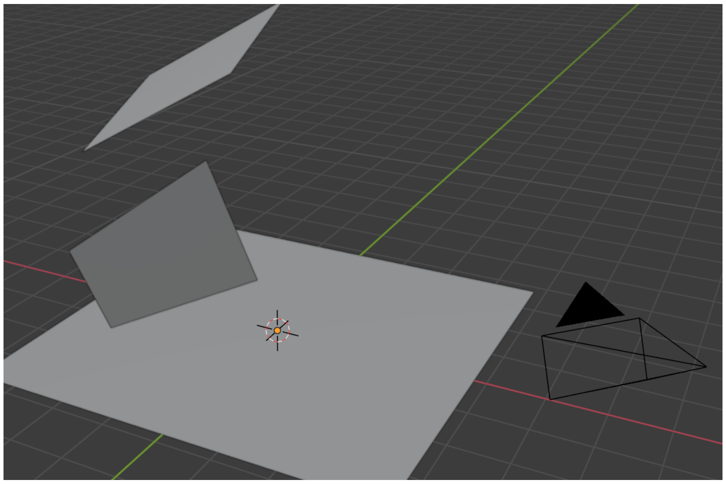
Figure 5: Collection of interesting photorealistic effects of the interaction of synthetic and virtual objects.

Exporting from Blender to PBRT

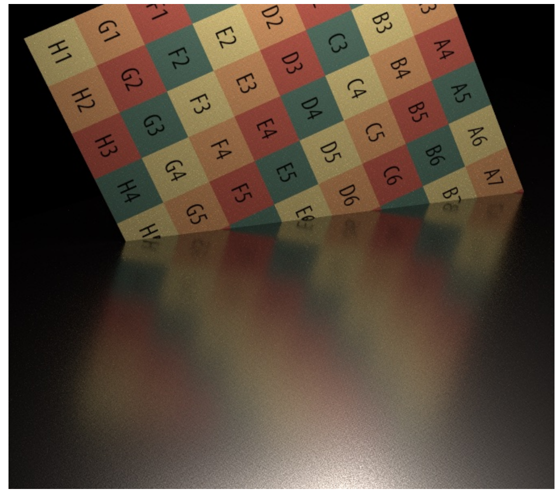
One of the steps in the workflow that might test your 'hacking' skills is the conversion of modelled scenes (in Blender) to a format that PBRT can be rendered with. Some tips:

1. Use a recent version of blender (I used 2.9x but 2.8x + should suffice).
2. Install the add-on from `https://github.com/stig-atle/io_scene_pbrt` and watch the instructional videos there.
3. When choosing materials for your modelled scene, using PBRT materials (important!).
4. Once you have modelled your scene, instead of rendering it in Blender, go to the render tab (as explained in the video) and click 'Export'.
5. Then render the exported .pbrt file using pbrt on the command line.

Some screenshots of a simple scene, and the exported .pbrt file are included below.



(a) Blender layout (light on top and camera on right).



(b) Output of PBRT.

```
Film "image" "integer xresolution" [1366]
  "integer yresolution" [768] "string filename" "output00001.exr"
PixelFilter "box" "float xwidth" [0.5] "float ywidth" [0.5]
Accelerator "bvh"
"string splitmethod" "sah"
"integer maxnodeprims" [4]
Sampler "halton"
"integer pixelsamples" [32]
"bool samplepixelcenter" "false"
Integrator "path"
"integer maxdepth" [10]
Scale -1 1 1
LookAt  7.358891487121582 -6.925790786743164 4.958309173583984
        6.707333087921143 -6.311620235443115 4.51303768157959
        -0.32401347160339355 0.305420845746994 0.8953956365585327

Camera "perspective"
"float fov" [22.89519204617112]
WorldBegin

        AttributeBegin
                Transform [1.363630 0.178212 0.773121 0.000000
                   0.348919 -1.515388 -0.266111 0.000000 0.712552
                   0.400999 -1.349234 0.000000 -1.761029 0.000000
                   4.542248 1.000000 ]
                AreaLightSource "diffuse" "blackbody L" [5500
                           10]
                Shape "trianglemesh"
                "point P" [
                -1.0 -1.0 0.0 1.0 -1.0 0.0 1.0 1.0 0.0 -1.0 -1.0 0.0
                   1.0 1.0 0.0 -1.0 1.0 0.0
```

```
            ]
            "normal N" [
            0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0
            ]
            "float st" [
            0.0 0.0
            1.0 0.0
            1.0 1.0
            0.0 0.0
            1.0 1.0
            0.0 1.0
            ]
            "integer indices" [
            0 1 2 3 4 5
            ]
    AttributeEnd


    AttributeBegin
            Transform [3.787295 0.000000 0.000000 0.000000 0.000000 3.787295 0.000000 0.000000 0.000000 0.000000 3.787295 0.000000 0.000000 0.000000 0.000000 1.000000 ]
            Material "metal"
            "color eta" [ 0.800000011920929 0.800000011920929 0.800000011920929]
            "color k" [ 0.800000011920929 0.800000011920929 0.800000011920929]
            "float roughness" [0.0]
            "float uroughness" [0.0]
            "float vroughness" [0.0]
            Shape "trianglemesh"
            "point P" [
            -1.0 -1.0 0.0 1.0 -1.0 0.0 1.0 1.0 0.0 -1.0 -1.0 0.0 1.0 1.0 0.0 -1.0 1.0 0.0
            ]
            "normal N" [
            0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0
            ]
            "float st" [
            0.0 0.0
            1.0 0.0
            1.0 1.0
            0.0 0.0
            1.0 1.0
```

```
                0.0 1.0
                ]
                "integer indices" [
                0 1 2 3 4 5
                ]
        AttributeEnd


        AttributeBegin
                Transform [0.394300 0.375732 -1.502911 0.000000
                   0.581884 1.401352 0.503004 0.000000 1.435733
                   -0.671138 0.208889 0.000000 -2.250519 0.000000
                   0.653642 1.000000 ]

                Texture "7frj09nru4zu.png" "color" "imagemap" "string
                   filename" ["textures/7frj09nru4zu.png"]
                Material "matte"
                "float sigma" [0.0]
                "texture Kd" "7frj09nru4zu.png"
                Shape "trianglemesh"
                "point P" [
                -1.0 -1.0 0.0 1.0 -1.0 0.0 1.0 1.0 0.0 -1.0 -1.0 0.0
                   1.0 1.0 0.0 -1.0 1.0 0.0
                ]
                "normal N" [
                0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0
                   0.0 1.0 0.0 0.0 1.0
                ]
                "float st" [
                0.0 0.0
                1.0 0.0
                1.0 1.0
                0.0 0.0
                1.0 1.0
                0.0 1.0
                ]
                "integer indices" [
                0 1 2 3 4 5
                ]
        AttributeEnd


WorldEnd
```