Instructions for Object Detection Training Image Creation

Table of Contents

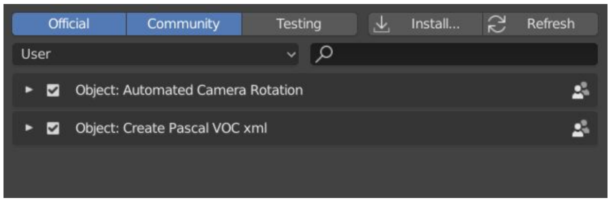
Installation

RevolveCamera.py & CreateXML.py

Both of these scripts are used in Blender and need to be installed as addons. If Blender is not installed, go to https://www.blender.org/download/ and download the latest release.

Install as Blender Addons:

- Go to Edit > Preferences > Add-ons
- Select "Install", select both RevolveCamera.py and CreateXML.py, then select "Install Add-on from File"
- Click their checkboxes to enable them.



▸ ☑ Object: Automated Camera Rotation

▸ ☑ Object: Create Pascal VOC xml

CreatePartialImage.pyw

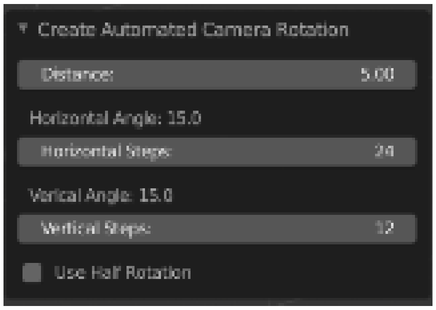
This is a standalone python script and requires Python 3.x to run. Python can be downloaded here: https://www.python.org/downloads/. To run this script, either double click the file or using the command prompt, navigate to the directory with the script and run it with `>python CreatePartialImage.pyw`

RevolveCamera.py

This script is used to create images from a 3D model in Blender. It will create a camera and animate it rotating around the object horizontally and vertically. This will result in anywhere from 1k - 5k images depending on the rotation angle and can take from 3 - 12 hours to render all of the images.

- Set up scene by placing the object at (0,0,0) and applying the desired material, and adding suitable lighting.
- Go to Object > Create Automated Camera Rotation.

- Modify parameters to desired value.
  - The number of steps is how many images it renders per rotation (more steps = more images, and fewer degrees in between each image)
  - To find a good camera distance, press num 0 to go to camera view and then alt-a to scrub through all frames without the parameters disappearing.
  - For symmetrical objects, check the “Half Rotation” box to render 180 degrees around the object.



Create Automated Camera Rotation

Distance: 5.00

Horizontal Angle: 15.0

Horizontal Steps: 24

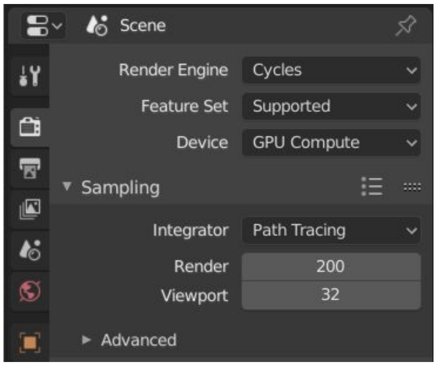
Vertical Angle: 15.0

Vertical Steps: 12

□ Use Half Rotation

Rendering

Rendering is not done with a script and is instead done with the tools that are already part of Blender. By default, the render settings are found on the right side of the screen under 3 tabs.



Scene

Render Engine: Cycles

Feature Set: Supported

Device: GPU Compute

Sampling

Integrator: Path Tracing

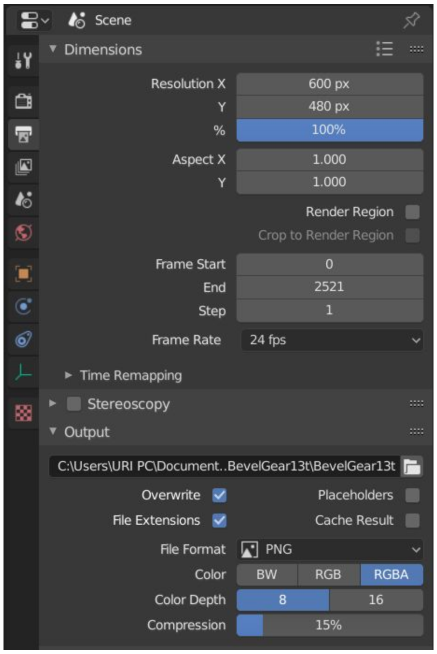
Render: 200

Viewport: 32

Advanced

- Render Settings tab
  - The Render Engine should be set to ‘Cycles’, and the device should be set to ‘GPU Compute’ if available. If ‘GPU Compute’ is not available, it may have to be enabled in Edit > Preferences > System > Cycles Render Devices.
  - The number of Render samples should be set anywhere from 120 to 250 (more samples produces a higher quality image but takes more time)
  - All other settings can be left at their default values.

- Output Settings tab
  - The resolution should be set to 600x480 at 100%.



Scene

Dimensions

Resolution X: 600 px

Y: 480 px

%: 100%

Aspect X: 1.000

Y: 1.000

Render Region □

Crop to Render Region □

Frame Start: 0

End: 2521

Step: 1

Frame Rate: 24 fps

Time Remapping

□ Stereoscopy

Output

C:\Users\URI PC\Document..BevelGear13t\BevelGear13t

Overwrite ☑ Placeholders □

File Extensions ☑ Cache Result □

File Format: PNG

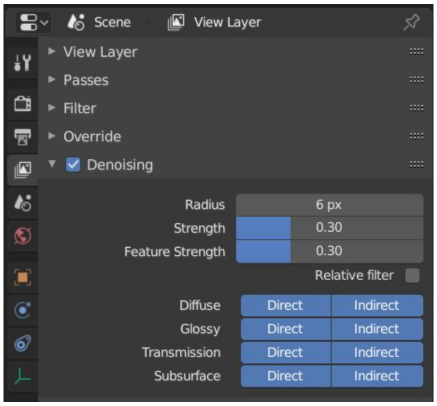
Color: BW RGB [✓ RGBA]

Color Depth: [✓ 8] 16

Compression: 15%

  - The output path should be set to an empty folder and the file name should be set appropriately. The File Format should be set to whatever format is needed to be input into the neural network.
  - All other settings can be left at their default values.

- View Layer Settings tab
  - Denoising should be enabled with a Radius of 6, a Strength of 0.30, and a Feature Strength of 0.30. These values can be changed depending on the desired output.
  - All other settings can be left at their default values.



☑ Denoising

Radius	6 px	
Strength	0.30	
Feature Strength	0.30	
Relative filter	□	
Diffuse	Direct	Indirect
Glossy	Direct	Indirect
Transmission	Direct	Indirect
Subsurface	Direct	Indirect

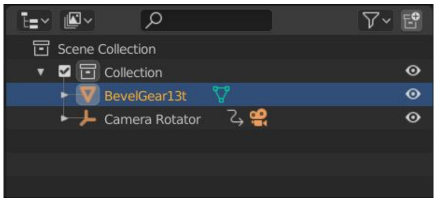
- To add a background to the scene, add a plane perpendicular to the camera behind the object. Apply the background material to the plane and parent the plane to the empty object created with the RevolveCamera script by selecting the plane, then the empty and pressing ctrl-p (set parent to object).

- After all settings have been properly adjusted, the images can be rendered by pressing ctrl-F12 or by going to Render > Render Animation. This process can be stopped prematurely and resumed by setting the ‘Frame Start‘ variable to the last frame rendered under the Render Settings tab.

CreateXML.py

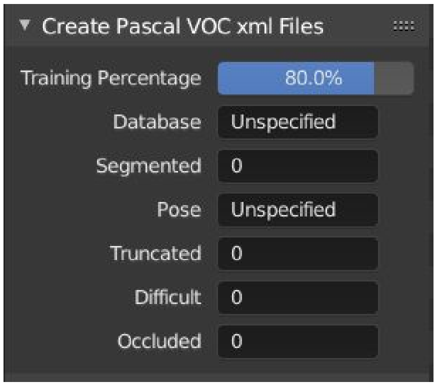
This script is used to create the xml files containing information and the bounding box for each image. It will create one xml file per image and will take from 10 seconds to 5 minutes depending on the amount of images and complexity of the model.

- Ensure that all of the images have been rendered to the desired folder and the object(s) you want to be included in the xml file is selected



in the scene. Hold shift while selecting to select multiple objects.

- The name of the object(s) in the xml file will be the same as the name of the object in the scene hierarchy.
- Go to Object > Create Pascal VOC xml Files.
- Navigate to the folder where the images have been rendered and make sure the file name is the same as the image name (e.g. gear13t, if the image names are gear13t####.jpg)
- In the bottom left, modify the parameters to the desired values.
    - "Training percentage" controls how many files are sorted into the training folder, and the remaining percentage are sorted into the testing folder.



Training Percentage 80.0%
Database Unspecified
Segmented 0
Pose Unspecified
Truncated 0
Difficult 0
Occluded 0

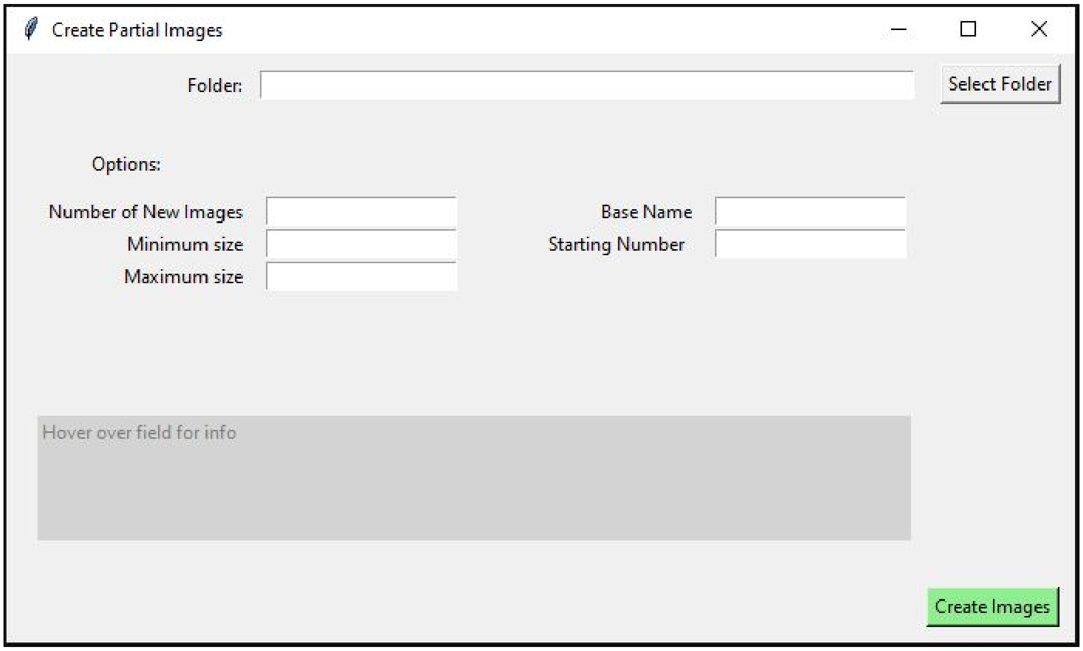
- The script will sort the images and xml files into "testing" and "training" folders. If these folders already exist, make sure they are empty. If they don't exist, the script will create them.
- Click "Create Pascal VOC xml Files". This will take anywhere from 10 seconds to 2 minutes depending on the number of images and complexity of the model.

CreatePartialImage.pyw

This script is used to create partial images from the original images and xml. All new images will have the same xml values of the base image, with the exception of the bounding box which will be recalculated hased on the new image. To use the script with the console open, change the .pyw extension to .py

- In the folder selection field, select the folder containing the images and xml files you want to create partial images with. If they are divided into 'training' and 'testing', you will have to run this program once with each folder.
- Modify parameters to the desired value.
    - Base Name is the name of the images without the file extension or the image number (e.g. *object* if the image names are *object1234.jpg*). This should be automatically filled in after the folder is selected.

    - Starting number should also be filled in automatically, and is the number in the naming scheme that the new images start at (e.g. *1235* if last image name is *object1234.jpg*).
    - Number of New Images is number of occluded images created from each original image.
    - Minimum size and Maximum size is the percentage size of the new images (e.g. 0.5 will result in an image half the size of the original image).
- Click the Create Images button the start creating the occluded images. The new images will be created in the original images are in and will take about 1 minute per thousand new images.



Author Information

Name: Joel White
University: Rochester Institute of Technology, Game Design and Development BS
Email: joelw@uri.edu or jxw5738@rit.edu