Generation Code for infinitexture: Unsupervised extraction of textures and PBR materials from images

1. Content:

1. **Extract_Textures.py:** Receives a folder of random images and extracts regions of uniform textures, crop and save to file.
2. **Turn_Texture_To_PBR.py:** Turn images of uniform textures (1) into PBR materials.
3. **Mix_PBRS.py:** Mix existing PBRS materials(2) to generate more diverse PBRs.

2. Extracting textures from images:

Identify regions with uniform textures in images, crop and save the textures as new images.

Script: **Extract_Textures.py:**

How to use:
Set input folder with random images to ***Image_dir***
Set output folders where the texture will be saved to ***out_dir***
The script should run out of the box with the sample data.
Extracted textures will appear **out_dir/large_textures**

Additional parameters:

Note the parameters below should be set depending on the image sizes you used and the quality and size of texture you want. The default value goes well with the segment_anything repository.

Tile_sizes = [75] : This defines the size of the cell in the texture the larger this parameter is the larger the extracted texture maps will be extracted. Note this is an array and can contain more than one cell size [70,80]... Good size for open_images repository is 40. Good size for segment-anything repository is 75.

***Min_tiles2map** = 7* : number of cells in texture across one dimension the larger this number the more uniform the textures will be and the larger they will be.

Sim_thresh = 0.5: How similar the region in textures should be the higher this number the higher the texture uniformity.

Note while it's possible to search for larger and more uniform textures by increasing the above parameters, this will also lead to less textures being found per image. The above value works good for segmenting anything repository for larger images you might increase them and vice versa.

3. Images source:

Any large folder of images can work for this the larger the images the larger the extracted textures can be. Good free source for large textures is the Segment_Anything_Image repository, good source for small images is the open_images repository.

4. Filtering smooth textures

Many of the textures extracted using **Extract_Textures.py** (2) are simply uniform colors or black or white regions, and for most applications should be removed.

Script: **Remove_Uniform_Regions.py** Identify and remove textures of uniform colors.

How to use:
Set input texture folder containing textures to be filtered to ***Image_dir.***
Set the output folder where the textures that were removed will be saved to ***out_dir.***
The script should run out of the box with the sample data.

5. Generation SVBRDF/PBR materials map from images

Generate PBR materials from a texture by using the various properties of the RGB texture to generate maps for various PBR properties (Roughness, normals, metallic).

Script: **Turn_Texture_To_PBR.py**

How to use:
Set input texture image folder into **texture_dir**
Set output folder where the output PBRs will be saved into **pbr_dir**
The script should run out of the box with the sample data.

Note: you can turn the PBR to seamless using: TurnPBR2Seamless.py, if you used seamless textures as input the PBRs will already be seamless.

6. Mixing SVBRDF/PBR materials

Another way to generate PBRs is by mixing existing PBRS to generate new ones.

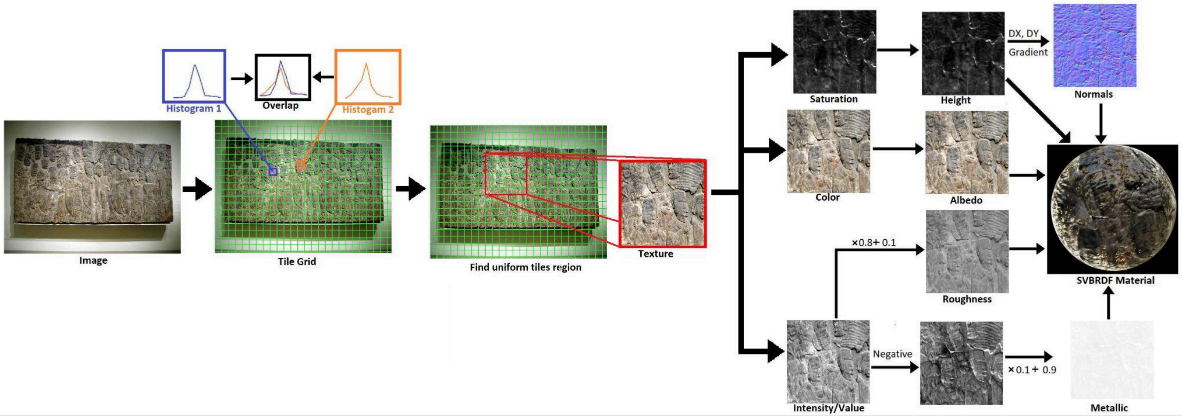
Script: **Mix_PBRS.py**

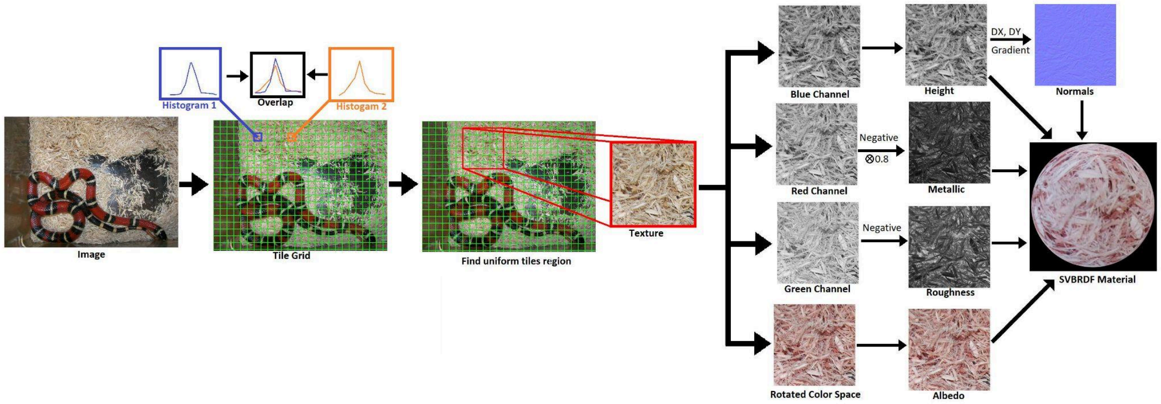
How to use:
Set the folder that contains PBR materials (each material as a subfolder) into **pbr_dir**
Set the output folder where the new mixed PBR will be saved in **merge_dir**
The script should run out of the box with the sample data.

How this works:

a) Divide the image into a grid. For every grid cell extract distribution of colors and gradients. Identify a region for which
all the cells have similar distributions as a uniform texture. Pick random channels from the extracted
texture image, augment them, and use the resulting maps as property maps (roughness, metallic,

height...) for the SVBRDF/PBR material.





License:

Code is available:
https://creativecommons.org/public-domain/cc0/

Images in sample images were taken from Segment_anything repository and available under segment anything dataset license.