SIG Instruction

Abstract

The Wheaton College Permanent Collection holds artwork ranging from the ancient Mediterranean world to contemporary American art. The project enables the public to browse 3D images of Wheaton's African art via website. We have created a streamlined process for generating 3D objects using photogrammetry, a technique that aids in the process of turning images into 3D models. The models are then uploaded to servers to be viewed online.

This instruction is modified by Yun Zhang '19 based on the original 'SIG Instructables' and 'Photogrammetry with A Drone' (obtained from https://souzoumaker.com/blog-1/2018/6/10/photogrammetry-with-a-drone) with permissions from both authors.

Hardware Assembly

Parts:

- High Torque Motor (200 rpm)
- Motor Speed Controller
- 12v Power Supply
- 15" Turntable
- Switches

The hardware used for this project is minimal, which allows anyone to begin scanning objects at a low cost. Our setup consists of a motorized turntable and a smartphone camera.

Assembly:

Print out all of the 3D model files. Cut and attach 2 sets of positive and ground wires to the motor speed controller input and output. Take the knob off the speed controller and slide it from under the housing and screw the nut from the other end to secure in place. Solder the output wires to the terminals of the motor. Attach the ground wire to the negative terminal on the 12v female plug. Solder the positive input wire to the side terminal of the switch. Make sure to pass the wire through the 3D print before soldering it to the switch (refer to image). Solder one more wire onto the switch (middle terminal) and attach to the positive terminal of the 12-female plug. Use hot glue to secure plug in place.
The turntable perimeter is fitted with a custom laser cut gear made out of wood. Attached perpendicular to the turntable gear is a gearbox housing the electronics and motor. Both laser cut gear and the bottom of the motor assembly are secured to the turntable with hot glue.

Attachments:

gear.stl
Download

Motor Holder.stl
Download

turntable attachment.stl
Download

turntable gear.pdf
Download

Generating Images

	Old Method (Total Images)	New Method (Video)	New Met (Generate Images)
e	596 MB ~ 150 images	150 MB ~ 24 sec	2.63 GB - images
	620 MB ~ 150 images	191 MB ~ 29 sec	2.36 GB - images
ad		162 MB ~ 26 sec	2.75 GB - images
nic		300 MB ~ 25 sec each (x2 videos)	3.83 GB - images to
ask		320 MB ~ 26 sec each (x2 videos)	4.28 GB - images to

Traditionally, individual images are taken by a camera of the object to be modeled; these images are then sent to the photogrammetry software to be rendered.

Our new method uses a video file as the source for images. A Python script using openCV is used to extract individual frames from the footage. The program allows for batch video processing, giving the user the flexibility to scan multiple objects in rapid succession. Several benefits are achieved as a result of this new technique including: reduction of hardware preparation, shorter time scanning objects, and data compression. In order to scan, we first place an object in the center of the turntable. Certain objects, such as necklaces, require slower rotation speeds to reduce vibrations that may cause the footage to blur. The camera is placed as close to the object as possible in order to capture the most detail. Videos from two or more angles may be required for larger pieces with sharper overhangs. Taking video of objects reduces the overall scanning time to a fraction of the previous method, as each video is approximately 30 seconds long. Digital photographs taken with a DSLR camera required much more preparation, as well as a person to take each photo manually. With the advent of video, storing object data has been improved. Images extracted using the Python program are only stored temporarily and can be discarded once the 3D model is generated. The old method requires storing individual images, increasing the required storage space as well as the challenge of file management. Generating

images from video has the added benefit of retaining greater detail of the scanned object, producing a higher quality 3D model. Above is a table of object file sizes.
The scripts for this program are available on GitHub for anyone to use: https://github.com/dihydrogenmonoxide21/VidCut.

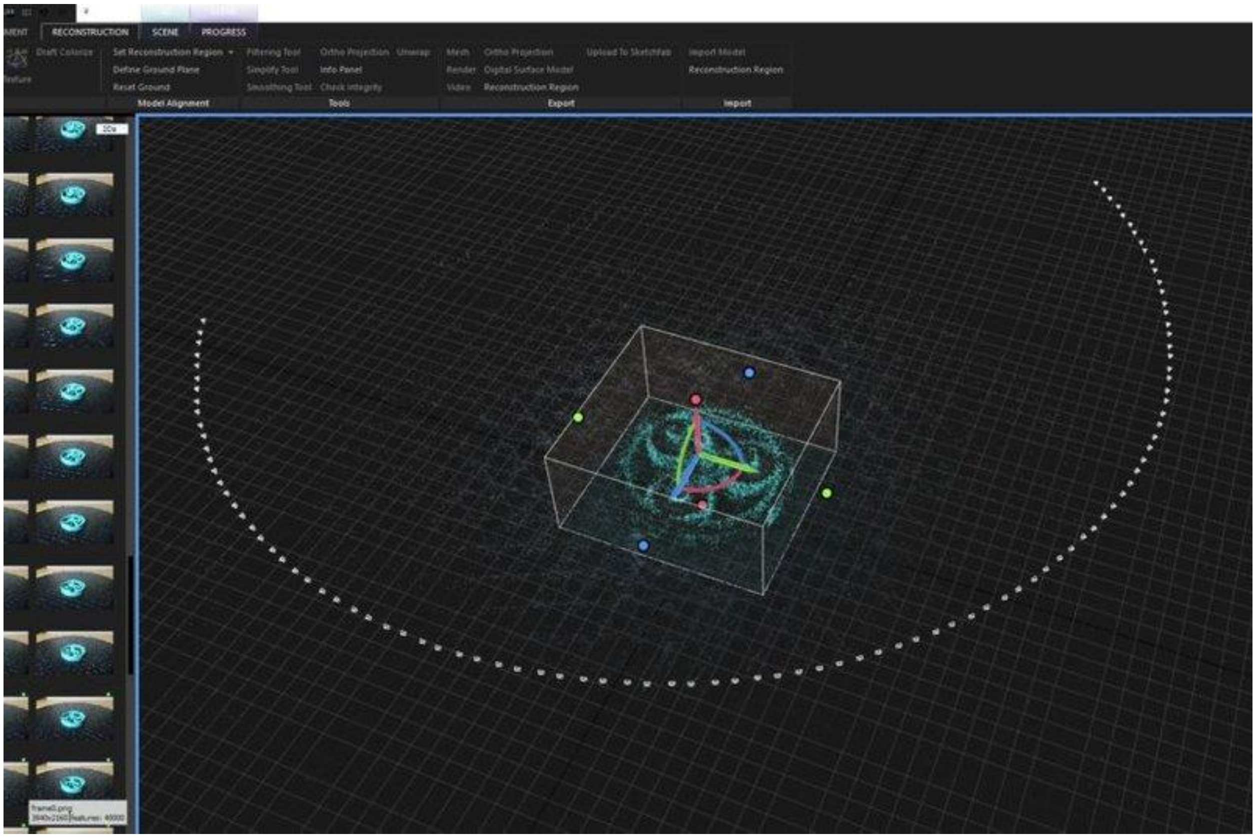
Running program on Windows:

Follow instructions here(https://github.com/BurntSushi/nfldb/wiki/Python-&-pip-Windows-installation) to install Python 2.7 and PIP to your computer. Once installed, open the command prompt and type "python" then run "pip install opencv-python" and "pip install rmtree". Use "cd" to navigate through files on your computer to where the python program is. (ex: "cd Desktop" if your python script is in the Desktop, or "cd Desktop\file\" to jump to a file in another file).
If running the single video processing program, make sure the video is in the same folder as the program. If running the batch video processing program, make a folder called "videos" and place all your videos in it. Type "python VidCutSingle.py" to run the program and wait to be prompted for the filename. Type in your video file (ex: test.mp4) with no double quotes - just the filename. Then the program will output the images into a new folder that is titled with the video filename. For the batch video program, make a folder called "videos" in the same file as the python file, and place all videos in that folder. Run via "python VidCutSingle.py" through the command prompt.

Running program on Mac OS:

Install python here (https://www.python.org/downloads/release/python-2715/). Open the terminal (found in utilities folder in applications) and type "sudo easy_install pip" in the new window. Once done, type "pip install opencv-python", then "pip install rmtree" to download the libraries that the program uses.
If running the single video processing program, make sure the video is in the same folder as the program. If running the batch video processing program, make a folder called "videos" and place all your videos in it. Type "python VidCutSingle.py" to run program and wait to be prompted for the filename. Type in your video file (ex: test.mp4) with no double quotes, just the filename. Then the program will output the images into a new folder that is titled with the video filename.
* The error message "ImportError: numpy.core.multiarray failed to import" may appear when the program is run. If this happens, enter the command "pip install -U numpy". This should remove the error message and the program should now execute properly.
For the batch video program make a folder called "videos" in the same file as the python file and place all videos in that folder. Run via the command "python VidCutSingle.py" through the terminal.

Photogrammetry



Once the images are prepared, they are now ready to be used to generate the 3D model. The photogrammetry software we use is called Reality Capture. Other photogrammetry programs, such as PhotoScan, would also work, but Reality Capture is by far the quickest to process. It goes through three steps to create the model: aligning images; rendering the 3D mesh; and rendering the texture.

Aligning:

Click on folder option to insert all generated images at once. Go to the alignment tab and select "align images". The software will align the images according to where it believes each photo was taken and in what orientation.
*For this example, only one pass of the turntable was used, however, a second pass of the turntable may be required for objects with sharper overhangs. It is recommended to take videos of objects from a variety of angles. Images from different angles can be added to the current process by simply selecting another folder.

Rendering the Mesh:

As a result of alignment, a point cloud of will be created. This is a three-dimensional representation of individual dots of features from the object that stand out. Rendering the mesh is like a 3D “connect the dots,” with the final product being a 3D model file. You may end up with a few stray dots in unexpected places, but these outliers are easy to eliminate. Under reconstruction, select "Set Reconstruction Region", and imagine making a rectangle using two opposite points from the ground plane of the model. Once setting the ground plane, move the mouse up to set the height. Even if it is not perfect, the region can be changed by dragging the colored dots on the face of the rectangle. Now, go to the reconstruction tab and select "normal detail" to generate the 3D model.

Texture:

In its current state, the 3D model file is a grayscale object. Select the texture option in the reconstruction tab to process the color. Once rendering is complete, the object can now be exported through mesh. This will come in the form of three files, the object file (3D model), the mtl file (material library), and the png file (texture for surface color).
On average, rendering takes 30 minutes using 250 images. Post processing of the 3D model is necessary in order for the website viewer to work. This is done by reducing the total triangles in the model within Reality Capture. Under the reconstruction tab, towards the middle, is the simplify tool. On average, 80,000 triangles seem to work best for compatibility with the online viewer.

<NOTE>: Filenames CANNOT be changed after those files are created. Changing filenames after saving step is completed will result in failure in load 3D models.

Website

Generated 3D models are available on our website for anyone to view. Viewing the models is made possible by the Online3DViewer by kovacsv, available on GitHub. To view our models, visit our website at: http://cs.wheatoncollege.edu/whedomains.