Photogrammetric Potsherd Profiling

Karl Edwards

2017-10-19

Abstract

An archaeologist lamented the fact that students having any interest in programming typically go off to study computer science. At the same time, students who are fully committed to archaeology have neither the interest nor the capacity to develop the types of computational tools that would speed up the tedious, repetetive aspects of archaeological practice. Perhaps specialists, in collaboration with generalists, can accomplish more together than either could accomplish alone. For example, imagine how useful it would be to leverage recent advances in photogrammetry and machine learning for the purpose of more quickly and accurately measuring a batch of objects and arranging them into a provisional hierarchy. This article describes a simple method for photographing pottery vessels and for turning those photographs into 3-D models, from which we derive characteristic measurements that will be useful in the evaluation of object similarity, and, ultimately, classification. We provide an example that includes fully-annotated **R** code for manipulating the modeled object in space, obtaining cross-sections, and calculating characteristic quantities. Subsequent articles will describe how to quantify similarity between objects, present a principled method for determining feature importance, and suggest an iterative process for generating provisional archetypes and resolving any resulting mis-classifications.

1. *Related Work*

Machine Learning for Archeology

A group of German researchers[1] describe how, using measurements obtained from a 3D-scanner, they have classified nearly 600 clay vessels from a Bronze-Age site in Saxony. Beginning with a pair-wise analysis of morphological features, they explain how to assess similarity between pairs of artifacts, and then find clusters of similar items. In the second phase, they develop a rational approach to calculating feature importance. Then they alternate between describing archetypes and classifying the artifacts using those preliminary definitions, adjusting the typology until the complete hierarchy emerges.

[1] Hörr, Lindinger, and Brunnett, in **Machine Learning Based Typology Development in Archaeology**

Photogrammetry Using Photoscan[2]

Björn K. Nilssen[3] combines photography with minimal manual modeling to create 3D models from of very large objects, such as sculptures and buildings The Poor Man's Guide To Photogrammetry[4] illustrates in great detail how to use Photoscan for modeling small objects.

[2] Photoscan is available at (http://www.agisoft.com)

[3] SketchUp with PhotoScan plugin (http://bknilssen.no/X/Photogrammetry/)

[4] Poor Man's Guide (http://bertrand-benoit.com/blog/the-poor-mans-guide-to-photogrammetry/)

Open Source Photogrammetry

For more control over the modeling process, the artist, Gleb Alexandrov, presents a photo scanning workflow based on open source projects VisualSFM[5], Meshlab[6], and Blender.[7]

[5] Visual Structure From Motion, by Changchang Wu(http://ccwu.me/vsfm/)

[6] Meshlab, the open source system for processing and editing 3D triangular meshes (http://www.meshlab.net)

[7] Blender, an open source 3D creation suite (https://www.blender.org)

Photogrammetry for Archeology

The nonprofic corporation, chi[8], demonstrates photogrammetry in the field of cultural heritage. The Archaeology Data Service at University of York has published a number of Guides to Good Practice, including Close-Range Photogrammetry: A Guide to Good Practice[9] Potsherd: Atlas of Roman Pottery[10] recommends CCD flatbed scanning for small, flat objects.

[8] Cultural Heritage Imaging

[9] Guides to Good Practice (http://guides.archaeologydataservice.ac.uk/g2gp/Phot 1)

[10] `http://potsherd.net/atlas`, P A Tyers, Sherd Scanning Advice (http://potsherd.net/atlas/topics/scanning)

2. Photography

You will need:

- One or more potsherds to measure
- Location with plenty of natural light. See photography guidelines
- Work table
- Camera
- Tripod
- Turntable
- Dark background

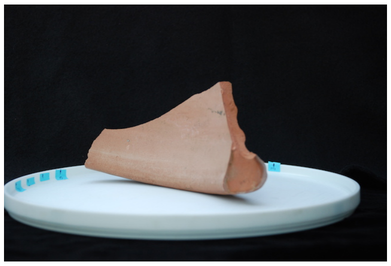


Figure 1: Turntable

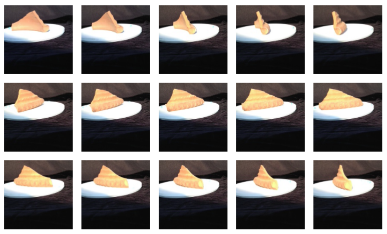


Figure 2: Turntable

Prepare the photography area as follows:

- Put dark paper or cloth under and behind the turntable to serve as a backdrop
- Adjust the height of the camera on the tripod so the center of the lense is at the same height as the center of the artifact to be photographed

Photograph each artifact

- Place the pot (or potsherd) on the turntable.
- Even though the accompanying photo does not illustrate this, prop the artifact so that the rim of the pot is parallel to the turntable (Rim up or rim down is not so important)
- To minimize distortion, place the artifact so that the center of the rim (or base) is approximately on the center of the turntable
- Take a series of photographs at roughly 10-degree intervals, resulting in 36 images per artifact

3. Model Creation

image files –> textured mesh –> stereolithography

The basic workflow is: (1) Take photographs from various perspectives, (2) Convert the photographs into a 3-Dimensional model, and, (3) since the model arrives as a textured mesh, convert it to stereolithography

Convert the image files to a textured mesh

- Upload images to ARC3D[11] web service

[11] If you are not already an ARC3D user, apply for a free account at `https://homes.esat.kuleuven.be/~visit3d/webservice/v2/request_login.php`

- Pour yourself a cup of coffee
- In a few minutes to a few hours, if all goes well, ARC3D will send you a textured mesh object

Convert the textured mesh to a stereolithography model

Enter the following command into a terminal window:

```
./meshconv textured_mesh.obj -c stl -o stereolithograph
```

This tells the conversion utility[12] three things: #. the object to use as input: *textured_mesh.obj*, #. the action(s) to perform: convert to stereolithography format [**-c stl**], #. where to put the results: in a file called **-o stereolithograph[.stl]**

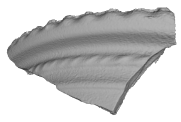


Figure 3: Textured Mesh Model

[12] The *meshconv* utility is available at (http://www.patrickmin.com/meshconv/)

4. Cross-Sectioning

The basic idea is to (1) find out which way is up, (2) orient the model squarely in the reference frame, (3) slice the model at various points between the base and rim, and, (4) slice the model from the perimeter toward the vertical axis of the vessel.

A. Model Orientation

Imagine you wish to put each potsherd in a box and you plan to stack hundreds of individually-packages sherds in your very small storage unit. If each box is a rectangular prism, how should you orient each sherd in order to minimize the volume of box required? One approach would be to make sure the base of each vessel is parallel to one side of its box.

To find the best orientation for the model of the sherd, we can take a similar approach. We begin by measuring the extent of the model along each axis, and calculating the product of these dimensions. Then we can tilt the model slightly, recalculate the volume, and repeat until we find the minimum.

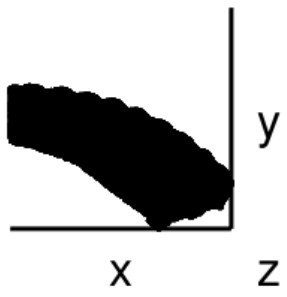
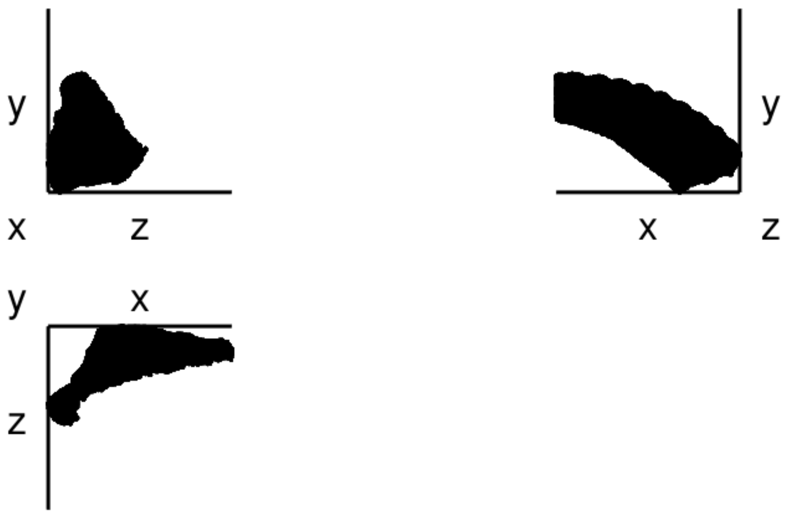


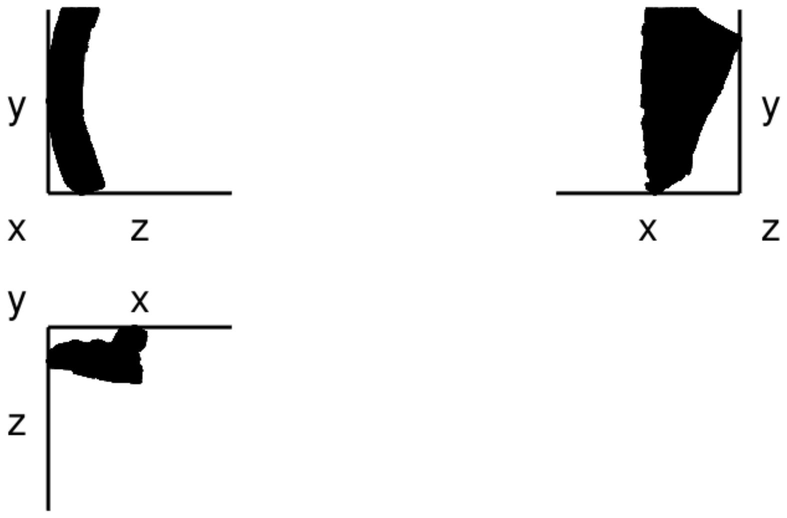
Figure 4: Initial Orientation

With the object oriented as shown below, the box size is 0.3653 units.



After repositioning the object, the box size is 0.1845 units.

```
multi_view( filenames_after )
```



B. Locate the Center of the Vessel

1. Slice the model horizontally
2. If the cross-section is roughly circular,

- repeat at various levels between base and rim

1. If the cross-section is *not* roughly circular,

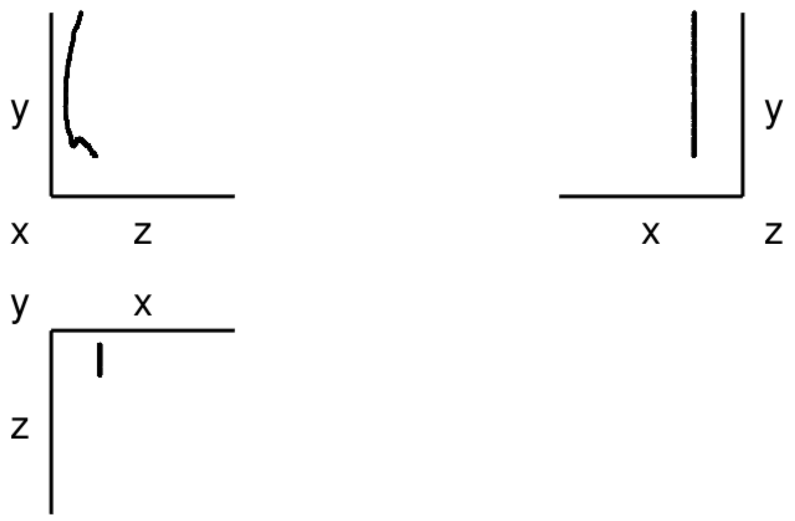
- choose each of the other axes in turn as the horizontal and select the best one.

```
# Duplicate the model, once for each axis
models <- list()
models <- map( 1:3, ~make_model( model$get() ))

# Slice through the middle three ways
# (once in each model)
mids <- model$calculate( f = get_midpoint )
walk( 1:3, ~models[[ . ]]$get_band( ax=., ctr = mids[ . ], thickness = 4*STRIPE_WIDTH ))
```

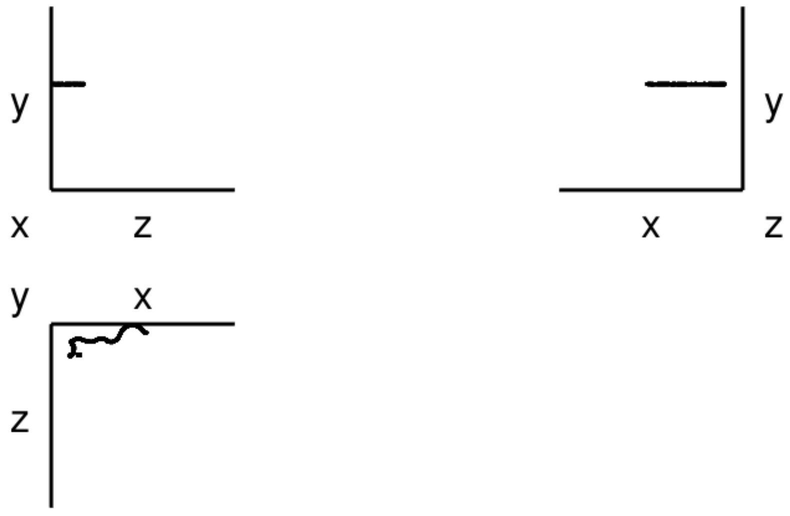
- Sliced on X

```
multi_view( filenames_slice_x )        # Show the slice from three perspectives
```



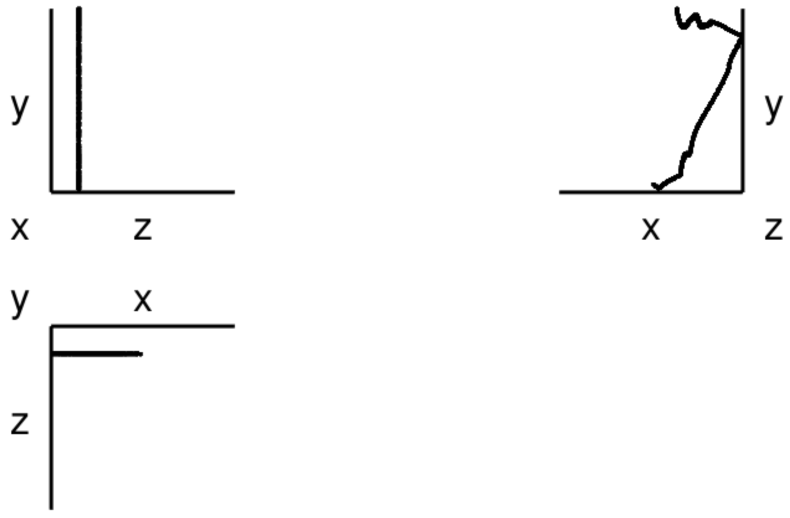
- Sliced on Y

```
multi_view( filenames_slice_y )        # Show the slice from three perspectives
```



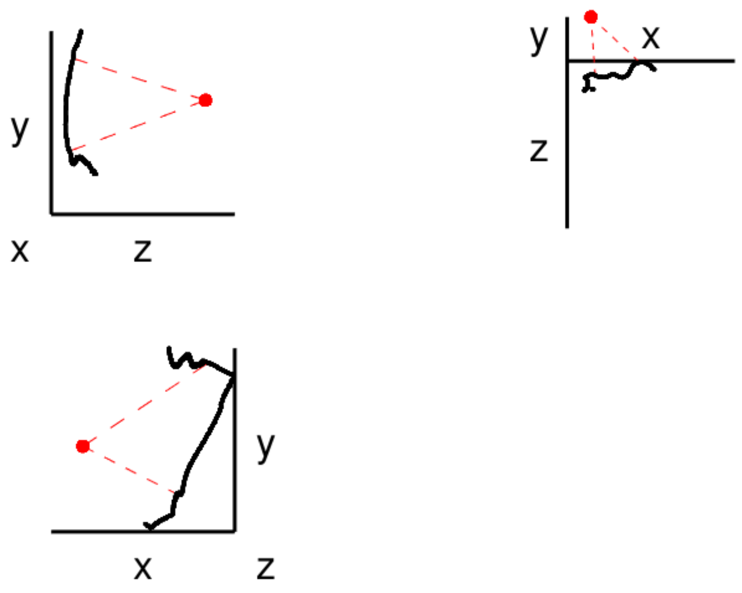
- Sliced on Z

```
multi_view( filenames_slice_z )        # Show the slice from three perspectives
```



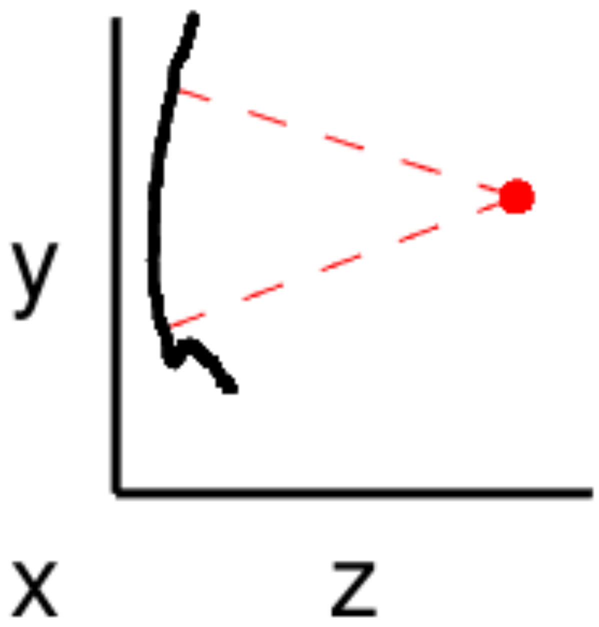
- With estimated centers

```
multi_view(                            # Show the three slices with estimated centers
  list( 'slice_x', 'slice_y', 'slice_z' )
)
```


y
x
z
y
x
z
y
x
z

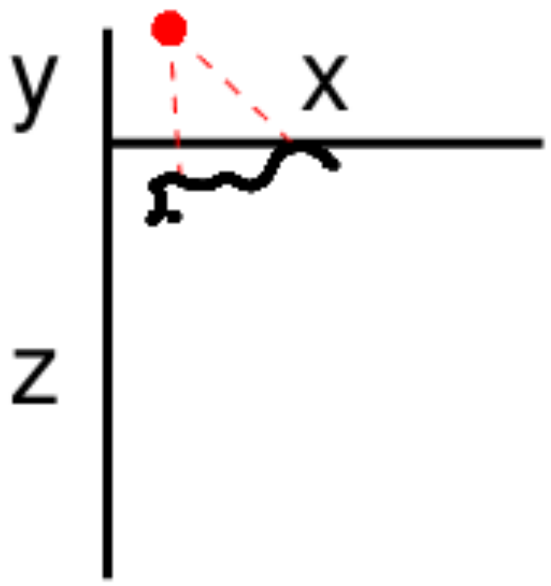
- X

```
grid::grid.raster( readPNG( paste0( FIGURES_PATH, 'slice_x.png' ) ) )
```



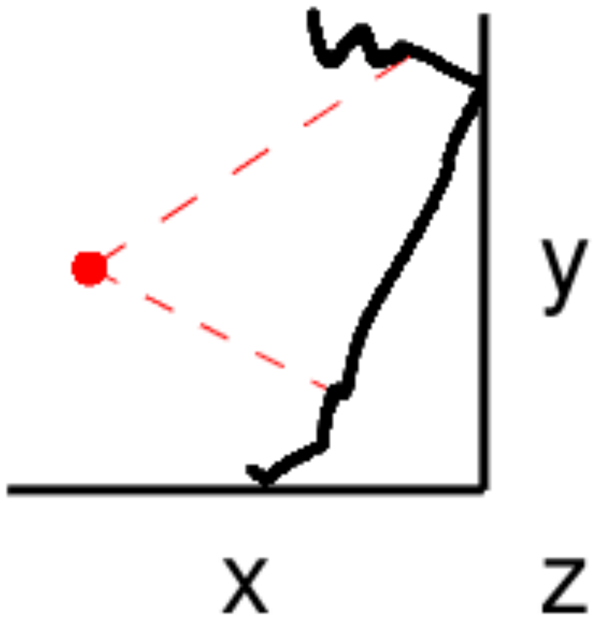
- Y

```
grid::grid.raster( readPNG( paste0( FIGURES_PATH, 'slice_y.png' ) ) )
```



- Z

```
grid::grid.raster( readPNG( paste0( FIGURES_PATH, 'slice_z.png' ) ) )
```



C. Profile

1. a numbered item
2. another numbered item

- a bullet-item
  - a sub-item
  - a sub-item

5. Measurement