

Kendra Bird  
SIE 510  
Final Project  
Spring 2014

## **Using Python to Calculate Spatial Locations of Cataloged Archaeological Artifacts**

### **Introduction and Objectives**

The purpose of this project was to correctly locate points associated with cataloged archaeological materials within a spatial context. The site referenced here is Holmes Point West, a Passamaquoddy shell midden site in Machiasport, Maine, which has been excavated in stages by the University of Maine's archaeological field school over several field seasons. The data utilized for this demonstration came primarily from the 2008, 2009, and 2010 excavations, although ultimately the application(s) developed as part of this project will be used to relocate catalog entries from 2012 and 2013 as well.

Spatial analysis of the artifact distributions at Holmes Point West is an important part of my research, and it is therefore essential that points intended to represent the locations of such materials be in the most ideal place possible. This ideal place varies based on the context of how the artifact was found. Holmes Point West is organized on a 1-meter-interval grid, and the coordinates of each pit are based on the north and east coordinates that correspond to the northeastern corner of each meter-by-meter square (e.g. N31E23). Each pit is broken down into four quads (identified based on their location; NW, NE, etc.), and each quad is typically vertically excavated in 5 cm. levels. Each level in each quad is assigned a unique provenience number (PN), and each artifact or group of artifacts cataloged within each PN is assigned an

accompanying catalog number (for example, if a PN was 1001, and the catalog number was 65, this would be formatted as such: 1001.65).

One of the problems that I initially encountered when attempting to place the artifacts in the catalog (represented by a Microsoft Access database) into a spatial context was that the only coordinates that I could assign to them were those north and east grid numbers. As a result, all points associated with an artifact from a particular pit would be assigned by default to the northeast corner of that pit. An artifact could have been found in the southwestern corner of the southwestern quad – as far away from the northeast corner as possible while still belonging to the same pit – and, as far as the as-is functionality of the catalog went, it would have been placed in that northeastern corner.

This would make it challenging, if not entirely impossible, to do any sort of useful analysis of the artifact placement and distribution at the site. One of the goals of my research is to identify activity areas in spatial, stratigraphic, and temporal contexts, expecting to see an inland, somewhat diagonal/horizontal, progression due to significant erosion at the site, which is not bedrock-defended in any way. In order to, at least in this context, extract any reasonable benefit from our dataset, I would have to find a more appropriate way of placing those points so that they more accurately reflected the location of the artifacts that they represented.

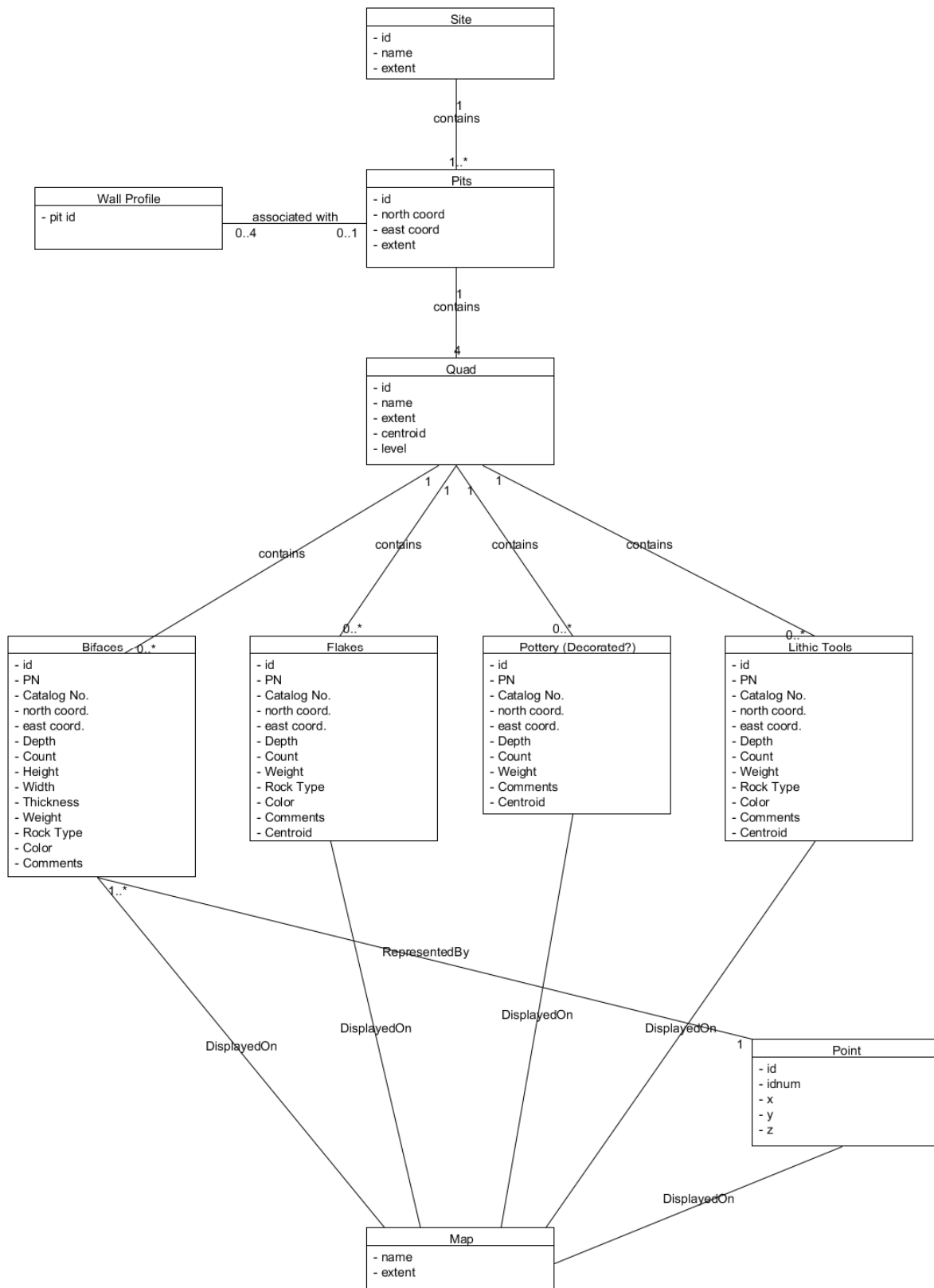
In order to do this, I planned to write a Python script which would extract information from a comma-delimited version of the catalog database, and then use that information to calculate the best-fit location for each point. The end result of this endeavor would be that the points were placed in the best possible spatial context, enabling me to visually represent distributions at the site and apply spatial analysis techniques to them.

## **Data Sources**

As mentioned earlier, the data utilized for this project is the result of cataloging artifacts retained from multiple seasons' worth of excavations at the Holmes Point West site. The data is contained in a Microsoft Access database file, which has been updated and revised over the years by a number of different individuals. Contained in a single table, the data was converted into a comma-delimited format for easy compatibility with my Python script. I then went back to the original table and extracted data pertaining to specific artifact types, creating new tables with which to simplify the joining process within ArcMap.

Total station measurements, collected by Samuel Belknap and Robert Ingraham in 2010, were used to construct a TIN file of the site, which is used on all accompanying maps to help show the context within which these points have been placed.

A conceptual model associated with this project can be found on the following page. It does not contain all of the attributes for each table, focusing instead on the most relevant ones in order to maintain simplicity. Ideally, the database will be streamlined considerably in the future, as it is not really designed in an efficient way; omitting some attributes is one way to present it in a more straightforward way without extensively revising it outright. Despite the omissions, this model should give a reasonable idea of how the components of the site, and, by extension, of this project, impact and are associated with one another.



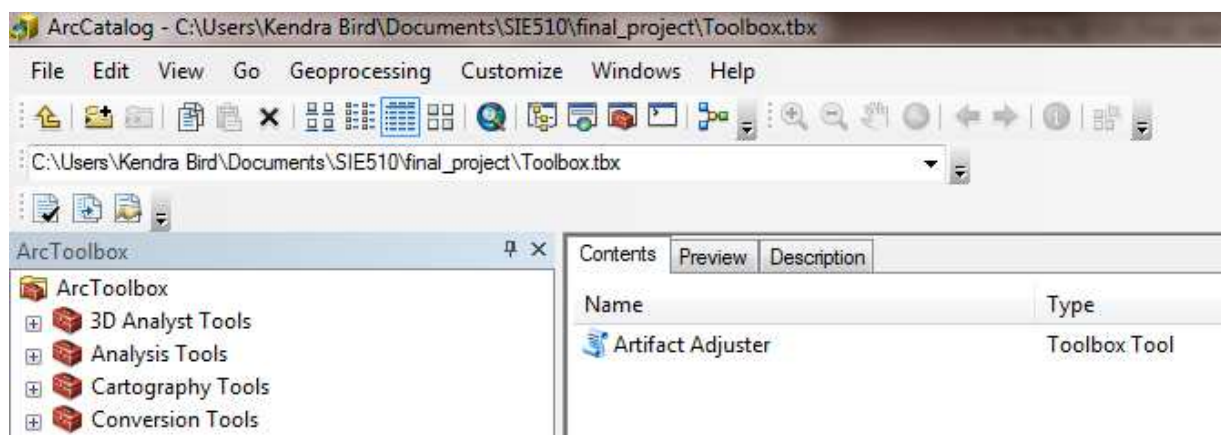
## **Methodology**

In order to achieve the objectives outlined earlier, I designed my Python tool to take a comma-delimited copy of the Holmes Point West catalog as input, break it into line segments in order to isolate needed information, and create a new point shapefile containing, among other information, revised X and Y coordinates corresponding to the ideal location for each point. Different contexts needed to be approached differently. In the interest of time and simplicity, I elected to generalize screen-found artifacts associated with features and column samples in the same way that other screen-found artifacts would be generalized. This meant that the appropriate calculations would be selected based on the quad associated with each entry in the catalog. If the context code corresponded to a screen find, feature, or column sample, then the next thing that would be checked would be the quad code. Based on that, the north and south coordinates, originally associated with those northeast corners, would be recalculated to correspond to the center of the appropriate quad, and then added, along with the corresponding ID, PN, and catalog number, to the output shapefile.

In situ artifacts, which are artifacts found in place in the field rather than in a screen, had to be handled differently. An in situ situation is better than a screen-find one, as it allows archaeologists to pinpoint exactly where, in both a horizontal and vertical context, an artifact was found, enabling me to reproduce its location in an exact way in models such as the one that I am working on for the Holmes Point West site. In the case of these artifacts, the Python script would identify the context code as corresponding to an in situ find, and then make its recalculations based on the specific measurements contained in the catalog that correspond to

the artifact's distance from the north and east walls. Again, these new coordinates and the accompanying information would be added to the new shapefile.

Transferring the ID number from each catalog entry enabled me to easily join the shapefile to the catalog tables (the primary one, plus the ones that I later created for specific artifact types), so the points, in their recalculated locations, would be connected to all relevant information and could be manipulated based on desired attributes, as well as analyzed in terms of quantities and whatnot.



As assigned, I converted my Python script into a custom tool, but it isn't set up to take any user input at the present time. However, in the future, I intend to revise it to take input at least in terms of desired output file path and name, and perhaps input file as well. As written, it is set up to handle a very specific set of circumstances, so I don't know how useful other user inputs would be.

I wrote an additional Python script (not converted into a tool) for use with this project, which constructed the pits corresponding to the 2008, 2009 and 2010 excavations as polygon shapefiles. It is a very simple script, and doesn't assign any identifying information to the pits aside from the year to which they belong, but it gave me a useful backdrop on which to place

my adjusted points. This script will also likely be worked on more in the future; as I wrote it, I did all the calculations myself for each point involved in each polygon, but those calculations are something that could certainly be automated based simply on the northeast coordinate as input. As I will have to use this again to generate pits to accompany the 2012 and 2013 seasons' data, I expect to make these revisions at some point over the summer.

Once processed, I used the new point shapefile representing the entire catalog to make new shapefiles based on artifact type, and then represented these in a variety of ways, such as quantity and density. I also isolated decorated pottery types, which are very useful in that different styles of decoration are associated with different time periods, and can help me map out the site in a temporal sense, not just a spatial one. Quantities are small enough for those that I couldn't do anything useful with the kernel or point density tools, but perhaps that will change when all of the data is entered into the catalog.

While all analysis at this point is very tentative, patterns corresponding to what we would expect to see are already evident based on the recalculated point shapefiles. For example, excavations suggested that there was a workshop area (based on the presence of significant quantities of flakes and bifaces) toward the eastern side of the site. Density analysis supports this idea; representing quantities as graduated symbols corresponds with this as well. Interestingly, the density of pottery in that area of the site is extremely low, while it is considerably stronger in the shell midden to the west. This helps support the idea of a workshop as well; discarded pottery might be found in a living space (which we have hypothesized to be represented below the shell midden in our site) or in a place for discarded, damaged materials, such as the midden itself.

The density of historic materials also corresponds roughly to the workshop area; in this region, pieces of a French pottery vessel have been found. While vertical or stratigraphic analysis has yet to take place in any significant way – and, indeed, Z coordinates haven't even been assigned to the historic materials yet – it seems reasonable to suggest that the workshop was potentially succeeded by some sort of short-term French presence during the contact period, and perhaps this part of the site was selected for use for a particular reason (visibility, solid ground, etc.). Something like a viewshed analysis might be useful for addressing the first idea, and identification of soils and stratigraphic layering might help with the other.

Maps illustrating the density and quantity analysis as well as the adjustment and distribution of various materials are included as an appendix to this paper.

At the very minimum, this project accomplished what it was intended to do: place the points associated with Holmes Point West artifacts into the best possible locations on the grid. Now that this has been accomplished, it opens the door for all sorts of additional analysis and generation of further questions.

### **Issues and Recommendations**

One of the biggest challenges with this project was the questionable quality of our database. I had to make a lot of revisions to make this work, as the attribute names were often formatted incorrectly, using special characters, spaces, or reserved words. These had to be changed to make the database work with ArcMap. Additionally, the person who worked most recently with the database prior to my tenure as lab manager set up the database to auto-format each cell regardless of input (i.e. it would capitalize a code even if it was typed in as all lower-case, or



mixed-case); while this simplifies things within Microsoft Access, the formatting unfortunately does not transfer into ArcMap, making it necessary to select for multiple codes in situations such as “Select by Attributes” when there should only be one. To separate out decorated pottery by type, I had to create a new field in the table and populate it with codes corresponding to each type; previously, the information about this was contained rather messily and inconsistently in the Description field. There are a lot of inconsistencies and flaws in our database (the fact that it was simply contained in one huge table is one), and it will take a lot of work to streamline; this will probably be one of my projects over the summer.

I also had some issues when joining the tables to the shapefiles; they joined easily enough, but in many cases I have multiple columns for the same attribute (CNT, CNT\_1, etc.). I assume this has something to do with the fact that I am joining a lot of tables with redundant information; this is something that should hopefully be rectified when I clean up the database.

Another data-related issue is that there are some problems with quads and with in situ measurements. I wrote my Python script in such a way that it will list the ID numbers for problem entries so I can go back and look at them (and, to be fair, most entries were handled successfully; the errors are few and far between). Quad issues are generally that an item came from one half of a pit but not a specific quad; in this case, I would add code to handle that by placing the point in the center of that rectangle. In situ issues typically are associated with cases in which range measurements were put in instead of a single measurement (e.g. an object is 22-24 cm. from the north wall and 80-83 cm. from the east wall). The way the script is written now is to only handle one number as input; a function checks to see if that particular line segment is actually a number, and rejects it (printing the corresponding ID number) if it is not. In these

instances, it interprets these hyphenated sequences as a string. One thing that might be possible would be to somehow create new line segments out of that particular line segment, using “-” as the delimiter; line segments 0 and 1 could then be averaged together to produce the number that would be multiplied by 0.01 and subtracted from the appropriate coordinate. In the interest of not breaking my currently-functional code, I opted not to test that out here, but I will try it in the future. I also need to revise my approach to feature and column samples, although their locations and dimensions vary so that will neither be an easy fix nor one that I think I possess the knowledge to successfully do at this point.

### **Conclusions**

Overall, this was a very successful endeavor. I had a fairly large dataset that was only marginally useful before, and I was able to turn it into something far more useful through the use of this Python script. As the dataset continues to grow, this tool will only increase in its usefulness, as it enables me to process the necessary data and generate spatially accurate, useful shapefiles in a matter of minutes. Additionally, the pit-making tool will grow in its usefulness as well, particularly if I implement the revisions that I intend to. I believe that my research has benefited greatly from the development of these tools, and I expect that I will use Python, especially in concert with ArcGIS, with even more frequency as my work continues.

**Appendix A: Python Code**

```
***** SIE 510 Final Project *****
***** Kendra Bird *****
***** Spring 2014 *****

# Import Arcpy
import arcpy

# Overwrite existing output file
arcpy.env.overwriteOutput = True

# Create point feature class (sealpath.shp) with the spatial reference 4269.
# Create fields for ID number, provenience number, catalog number, adjusted_X and
adjusted_Y.
arcpy.CreateFeatureclass_management("C:/users/kendra
bird/documents/sie510/final_project/", "holmes_point_catalog", "POINT",
spatial_reference=4269)
outshapefile = "C:/users/kendra
bird/documents/sie510/final_project/holmes_point_catalog.shp"
arcpy.AddField_management(outshapefile, "IDnum", "LONG", field_precision=5)
arcpy.AddField_management(outshapefile, "PN", "TEXT", field_length=4)
arcpy.AddField_management(outshapefile, "Cat", "TEXT", field_length=2)
arcpy.AddField_management(outshapefile, "adjusted_X", "FLOAT", field_precision=6,
field_scale=4)
arcpy.AddField_management(outshapefile, "adjusted_Y", "FLOAT", field_precision=6,
field_scale=4)

# Initialize count variables for use in loops
bad_quad = 0
bad_meas = 0
lines = 0

# Function to check if line segment is a number
def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        return False

# Open Holmes Point Catalog file and create insert cursor
with open("C:\\Users\\Kendra Bird\\Documents\\SIE510\\final_project\\Holmes Point
Catalog.csv") as file:
```

```
icur = arcpy.da.InsertCursor(outshapefile, ("IDnum", "PN", "Cat", "adjusted_X", "adjusted_Y",
"SHAPE@XY"))
```

```
# Read and ignore header
header1 = file.readline()
```

```
# Split csv file into segments, assigning variables and types to specific segments
```

```
for line in file:
    segmentedline = line.split(",")
    ID = int(segmentedline[0])
    PN = str(segmentedline[1])
    Cat = str(segmentedline[2])
    y = float(segmentedline[3])
    x = float(segmentedline[4])
    quad = str(segmentedline[5])
    context = str(segmentedline[16])
    nWall = str(segmentedline[7])
    eWall = str(segmentedline[8])
```

```
# Identify, calculate, and process locations for in situ artifacts based on context code, taking
note of incorrectly-formatted (non-number) entries
```

```
if (context == "IS") or (context == "is") or (context == "iS"):
    true = is_number(segmentedline[7])
    if true:
        nWall = (float(segmentedline[7]))
        true = is_number(segmentedline[8])
        if true:
            eWall = (float(segmentedline[8]))
            y = y-(((nWall)*0.01))
            x = x-(((eWall)*0.01))
            row = (ID, PN, Cat, x, y, (x,y))
            icur.insertRow(row)
            lines = lines + 1
        else:
            print "Incorrectly formatted entry: %s" %(segmentedline[0])
            continue
```

```
true = is_number(segmentedline[8])
if true:
    eWall = (float(segmentedline[8]))
    true = is_number(segmentedline[7])
    if true:
        nWall = (float(segmentedline[7]))
        y = y-(((nWall)*0.01))
        x = x-(((eWall)*0.01))
```

```

        row = (ID, PN, Cat, x, y, (x,y))
        icur.insertRow(row)
        lines = lines + 1
    else:
        print "Incorrectly formatted entry: %s" %(segmentedline[0])
        continue

# Identify, calculate, and process locations for screen find, column sample, and feature artifacts
based on context code, taking note of bad quads
    elif (context == "SCFD") or (context == "scfd") or (context == "COLM") or (context == "colm")
or (context == "FEAS") or (context == "feas") or (context == "SCRM"):
        if (quad == "SE") or (quad == "se"):
            y = y-0.75
            x = x-0.25
        elif (quad == "SW") or (quad == "sw"):
            y = y-0.75
            x = x-0.75
        elif (quad == "NE") or (quad == "ne"):
            y = y-0.25
            x = x-0.25
        elif (quad == "NW") or (quad == "nw"):
            y = y-0.25
            x = x-0.75
    else:
        bad_quad = bad_quad + 1
        lines = lines + 1
        print "Bad quad: %s" %(segmentedline[0])
        continue
    row = (ID, PN, Cat, x, y, (x,y))
    icur.insertRow(row)
    lines = lines + 1
    continue
else:
    lines = lines + 1
    bad_meas = bad_meas + 1
    print "Incorrect context: %s" %(segmentedline[0])
del icur, row

# Print totals of coordinate pairs (processed, skipped, total)
print "Processing complete."
print "Total number of entries processed: %i" %(lines)
print "Total number of entries skipped due to incorrect context: %i" %(bad_meas)
print "Total number of bad quads: %i" %(bad_quad)

```

```
# Close the file
file.close
```

---

```
##### Pit Maker #####
##### SIE 510 #####
##### Kendra Bird #####
```

```
# import arcpy
import arcpy
```

```
# Overwrite existing output file
arcpy.env.overwriteOutput = True
```

```
# list that will hold each polygon
pits08 = []
pits09 = []
pits10 = []
```

```
# list of coordinates used to generate polygons
coordList08 =
[[[26,14],[25,14],[25,13],[26,13],[27,14],[26,14],[26,13],[27,13],[20,20],[19,20],[19,19],[20,19]],
[[24,22],[23,22],[23,21],[24,21]],
[[21,26],[20,26],[20,25],[21,25],[23,26],[22,26],[22,25],[23,25],[27,26],[26,26],[26,25],[27,25]],
[[28,26],[27,26],[27,25],[28,25]],
[[29,26],[28,26],[28,25],[29,25],[22,29],[21,29],[21,28],[22,28],[20,29],[19,29],[19,28],[20,28]],
[[21,38],[20,38],[20,37],[21,37]],
[[19,31],[18,31],[18,30],[19,30],[31,52],[30,52],[30,51],[31,51],[31,53],[30,53],[30,52],[31,52]],
[[20,27],[19,27],[19,26],[20,26]]]
```

```
coordList09 =
[[[27,23],[26,23],[26,22],[27,22],[25,22],[24,22],[24,21],[25,21],[25,23],[24,23],[24,22],[25,22]],
[[23,24],[22,24],[22,23],[23,23]],
[[18,25],[17,25],[17,24],[18,24],[19,25],[18,25],[18,24],[19,24],[20,25],[19,25],[19,24],[20,24]],
[[27,25],[26,25],[26,24],[27,24]],
[[28,25],[27,25],[27,24],[28,24],[27,26],[26,26],[26,25],[27,25],[26,27],[25,27],[25,26],[26,26]],
[[28,26],[27,26],[27,25],[28,25]],
[[24,27],[23,27],[23,26],[24,26],[27,27],[26,27],[26,26],[27,26],[28,27],[27,27],[27,26],[28,26]],
[[29,27],[28,27],[28,26],[29,26]],
[[30,27],[29,27],[29,26],[30,26],[26,28],[25,28],[25,27],[26,27],[27,28],[26,28],[26,27],[27,27]],
[[28,28],[27,28],[27,27],[28,27]],
[[23,31],[22,31],[22,30],[23,30],[29,47],[28,47],[28,46],[29,46]]]
```

```

coordList10 =
[[[29,17],[28,17],[28,16],[29,16]],[[30,24],[29,24],[29,23],[30,23]],[[29,25],[28,25],[28,24],[29,24]],[[30,25],[29,25],[29,24],[30,24]],
[[32,25],[31,25],[31,24],[32,24]],[[29,26],[28,26],[28,25],[29,25]],[[30,26],[29,26],[29,25],[30,25]],[[31,27],[30,27],[30,26],[31,26]],
[[32,27],[31,27],[31,26],[32,26]],[[20,28],[19,28],[19,27],[20,27]],[[29,28],[28,28],[28,27],[29,27]],[[30,28],[29,28],[29,27],[30,27]],
[[26,29],[25,29],[25,28],[26,28]],[[28,29],[27,29],[27,28],[28,28]],[[24,30],[23,30],[23,29],[24,29]],[[24,31],[23,31],[23,30],[24,30]],
[[17,36],[16,36],[16,35],[17,35]],[[23,30],[22,30],[22,29],[23,29]],[[19,28],[18,28],[18,27],[19,27]]]]

```

```
# initialize count variable
```

```
total = 0
```

```
# Loop to append coordinates to 2008 polygon shapefile
```

```
for pit in coordList08:
```

```
    pits08.append(
```

```
        arcpy.Polygon(
```

```
            arcpy.Array([arcpy.Point(*coords) for coords in pit]))
```

```
    total = total + 1
```

```
# print results
```

```
print "2008 Processing complete."
```

```
print "%i pits were created."%(total)
```

```
# save array to shapefile
```

```
arcpy.CopyFeatures_management(pits08, "C:/users/kendra
bird/documents/SIE510/final_project/2008pits.shp")
```

```
# initialize count variable
```

```
total = 0
```

```
# Loop to append coordinates to 2009 polygon shapefile
```

```
for pit in coordList09:
```

```
    pits09.append(
```

```
        arcpy.Polygon(
```

```
            arcpy.Array([arcpy.Point(*coords) for coords in pit]))
```

```
    total = total + 1
```

```
# print results
```

```
print "2009 Processing complete."
```

```
print "%i pits were created."%(total)
```

```
# save array to shapefile
arcpy.CopyFeatures_management(pits09, "C:/users/kendra
bird/documents/SIE510/final_project/2009pits.shp")

# initialize count variable
total = 0

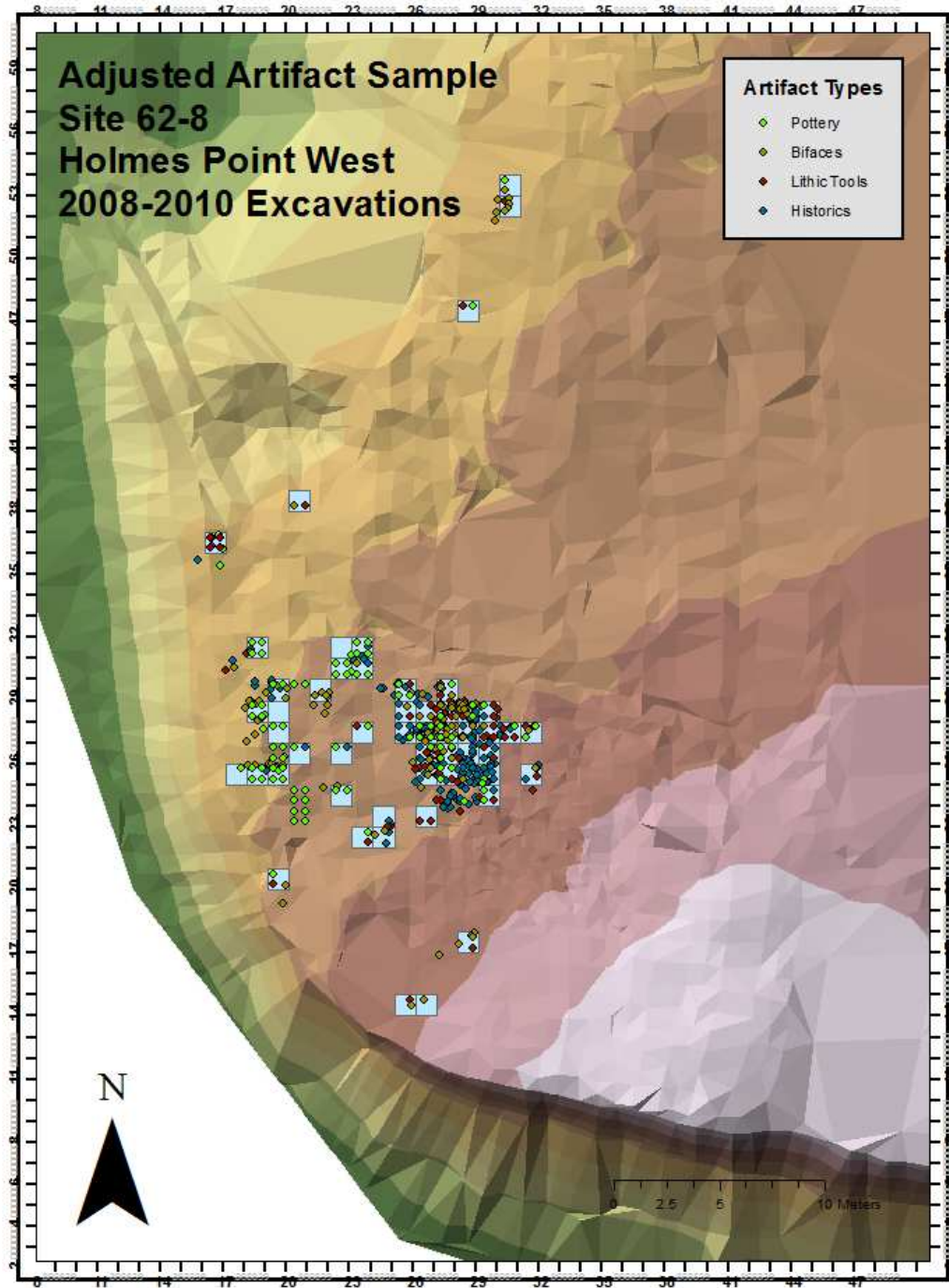
# Loop to append coordinates to 2010 polygon shapefile
for pit in coordList10:
    pits10.append(
        arcpy.Polygon(
            arcpy.Array([arcpy.Point(*coords) for coords in pit])))
    total = total + 1

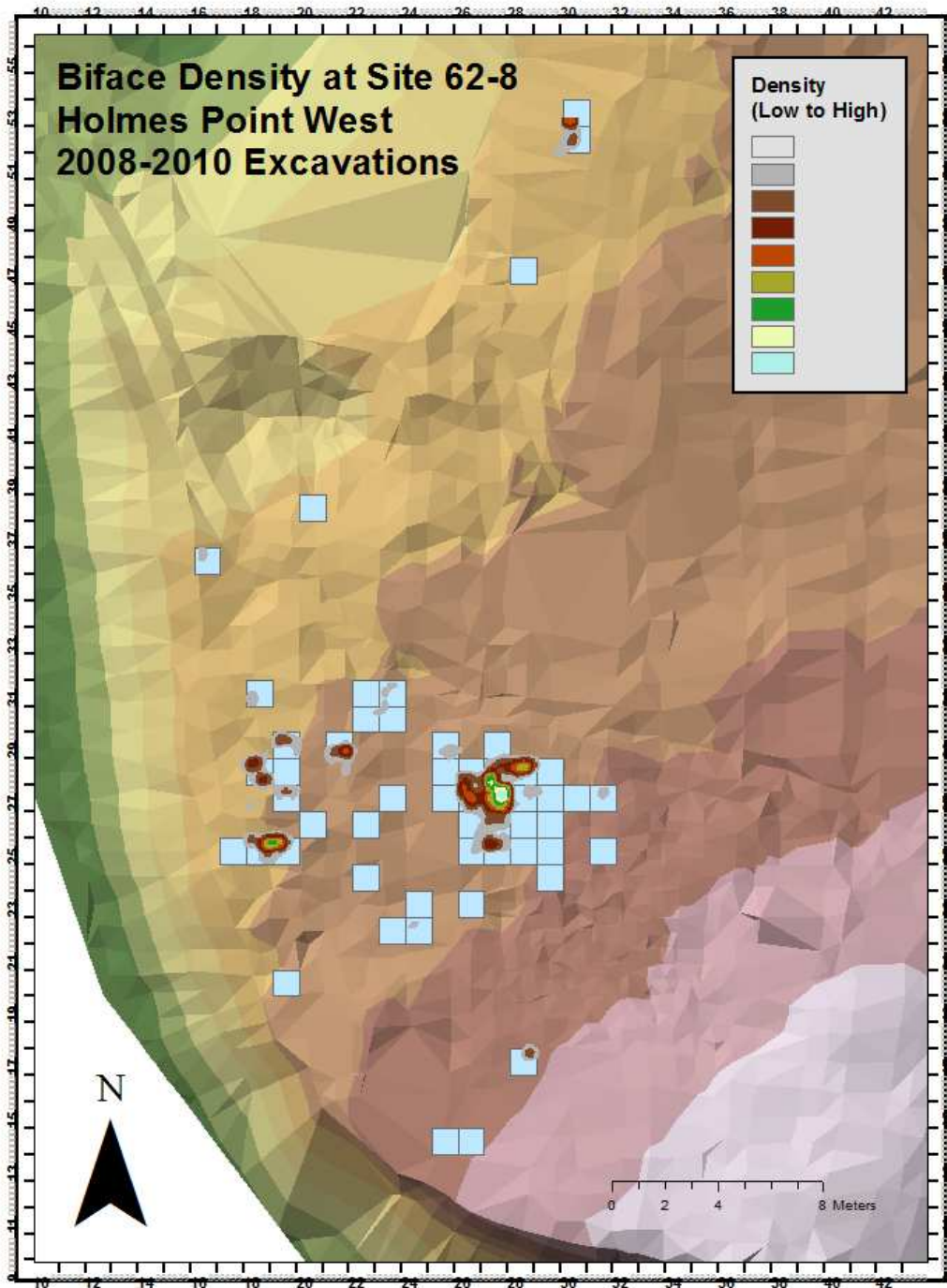
# print results
print "2010 Processing complete."
print "%i pits were created."%(total)

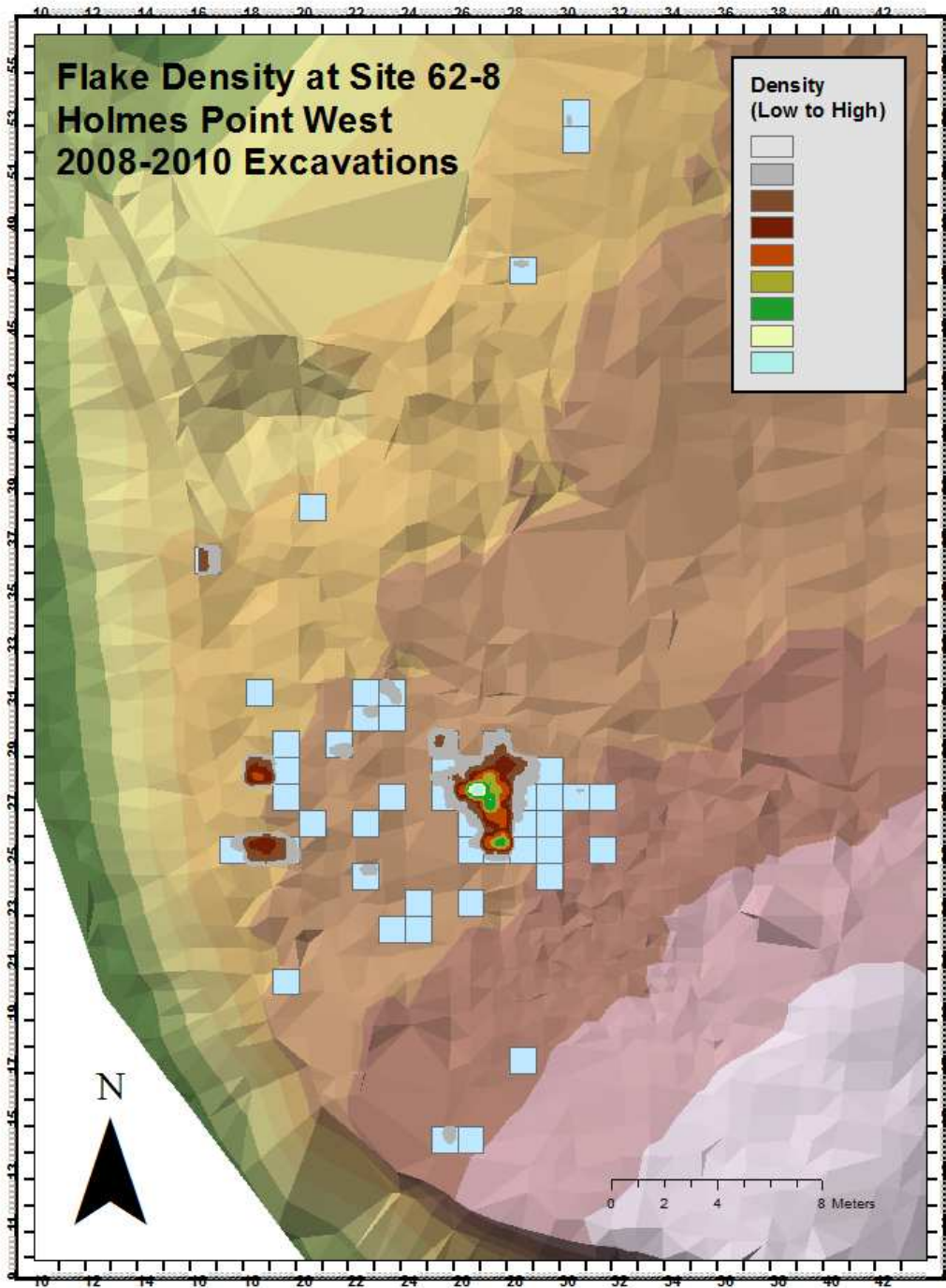
# save array to shapefile
arcpy.CopyFeatures_management(pits10, "C:/users/kendra
bird/documents/SIE510/final_project/2010pits.shp")
```

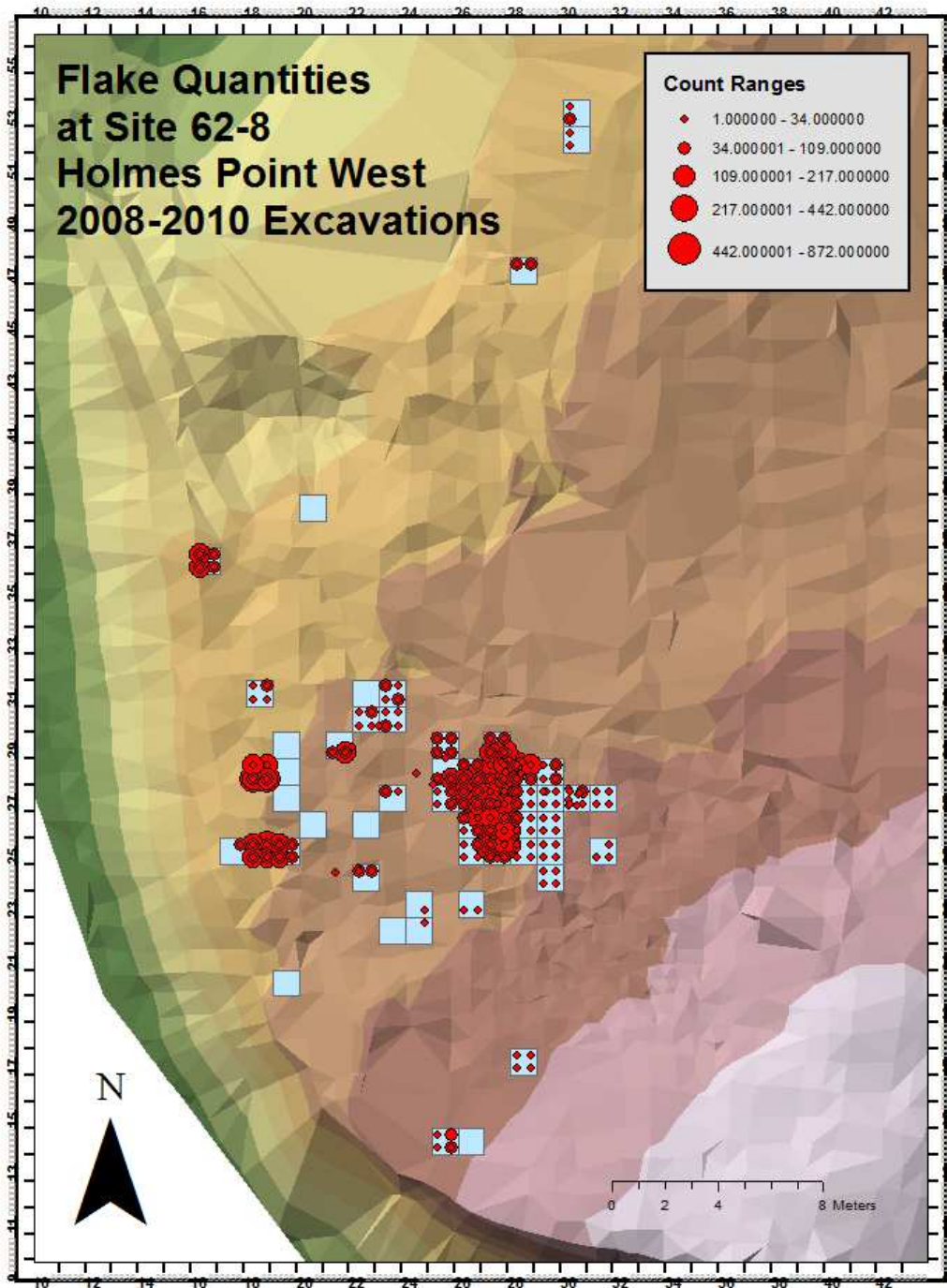


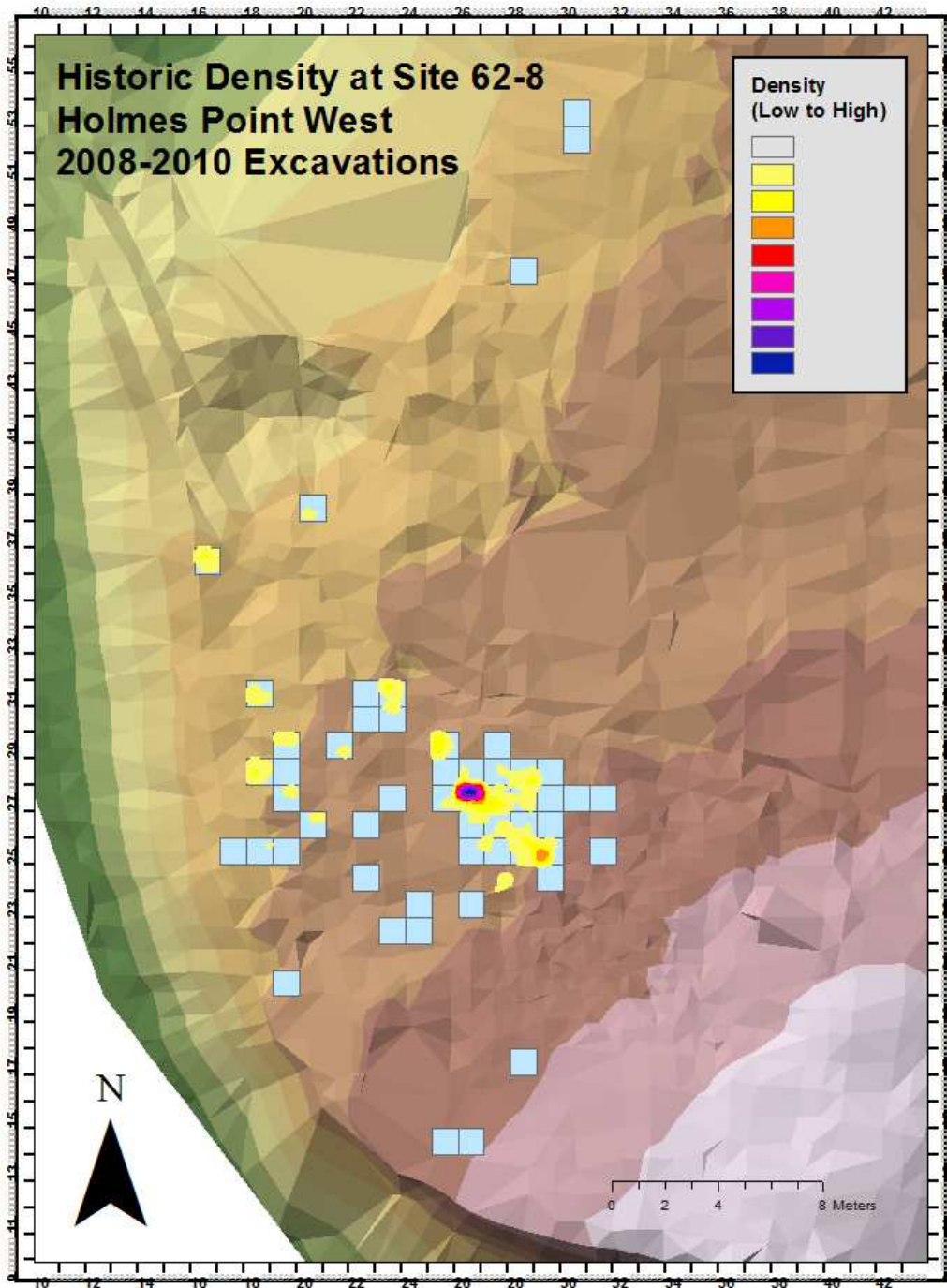
Appendix B: Maps

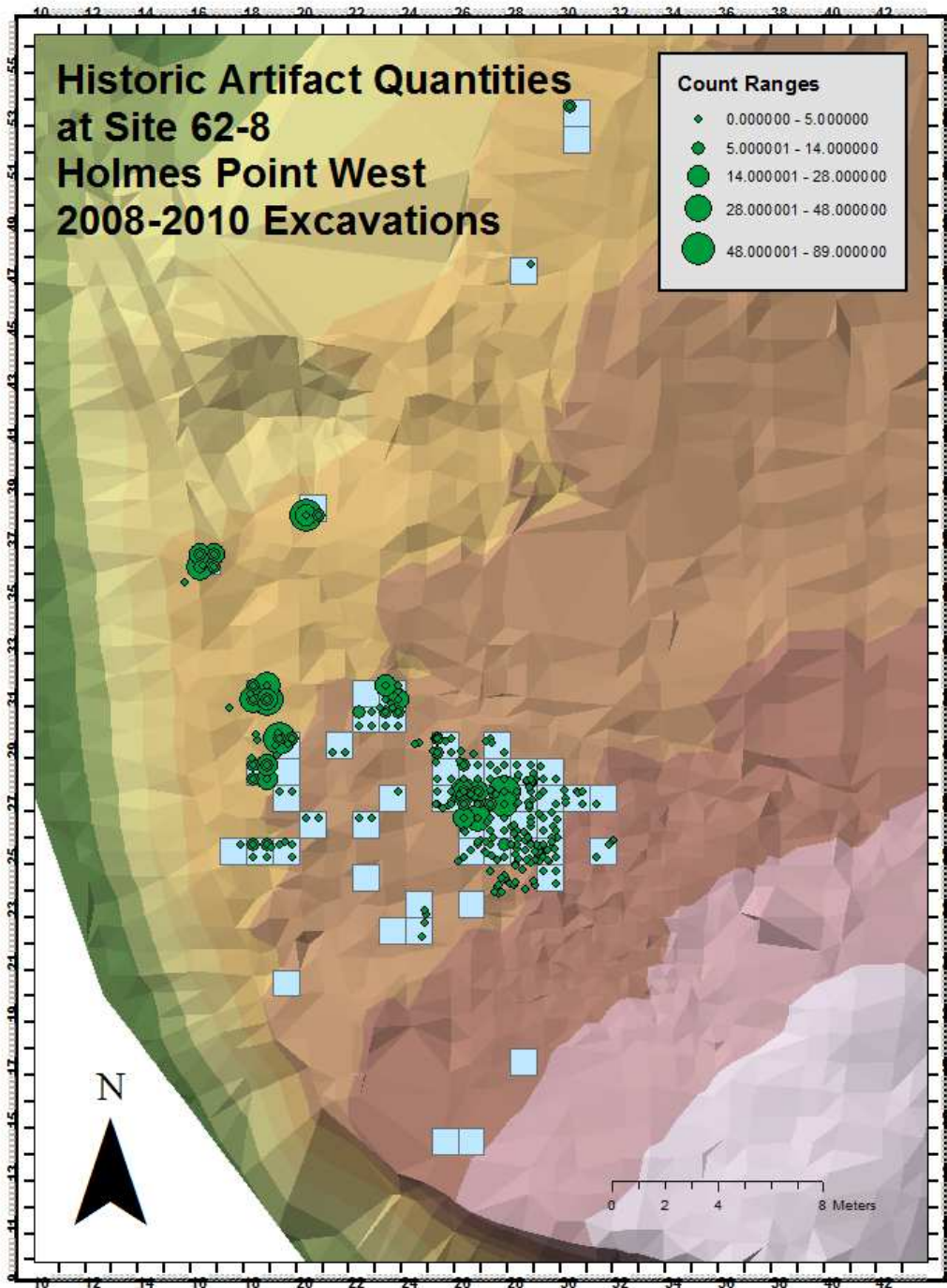


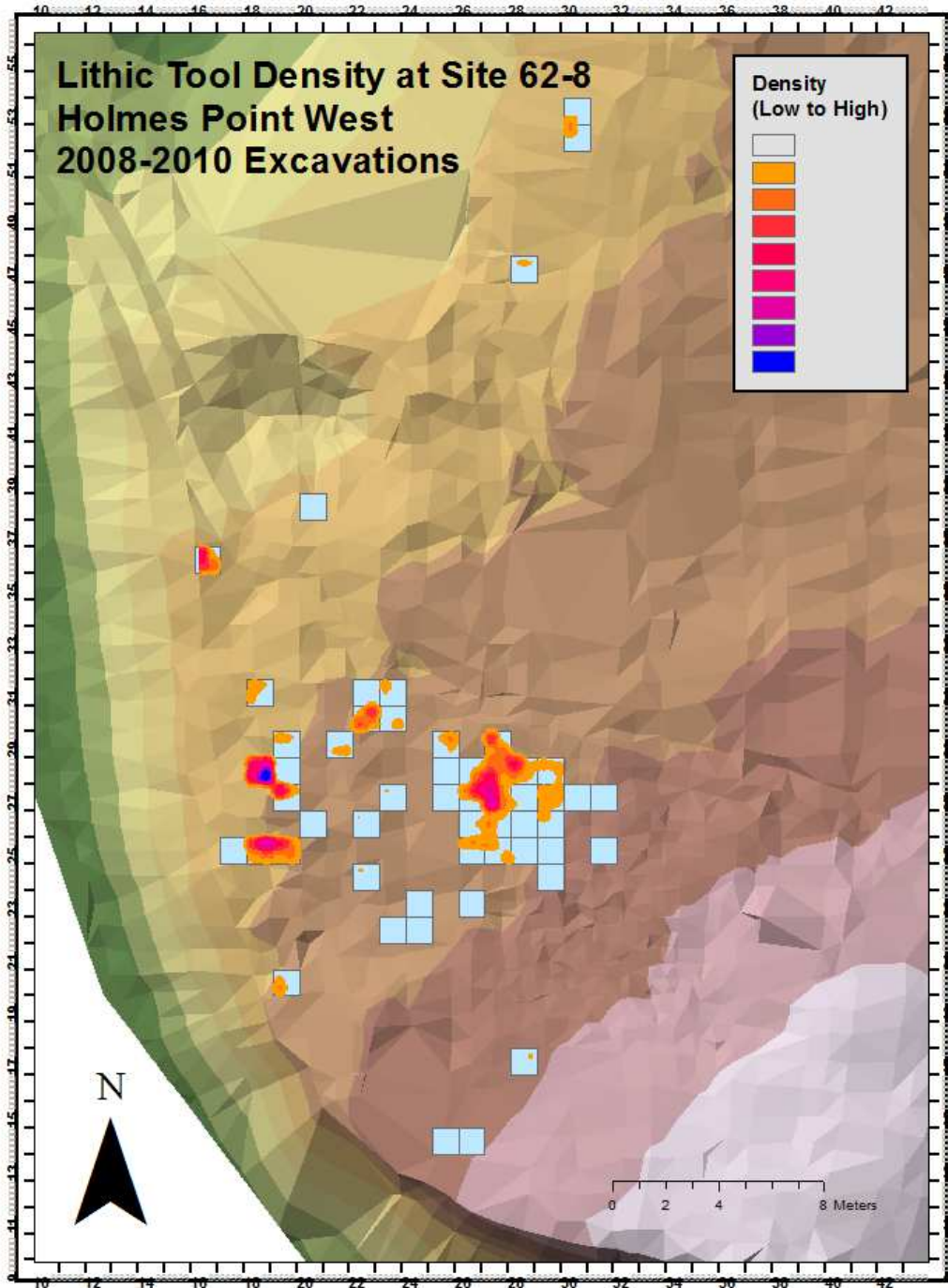


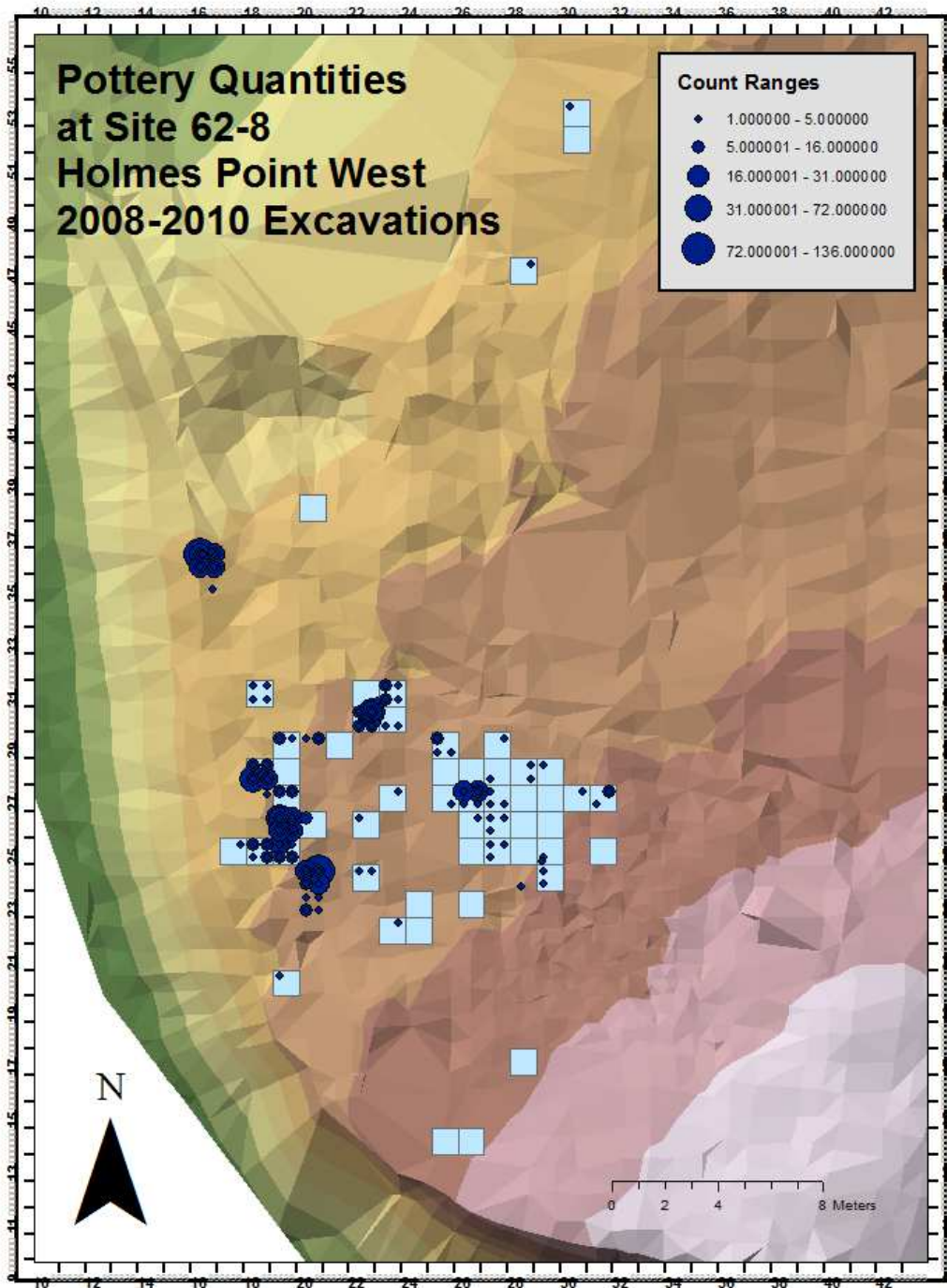




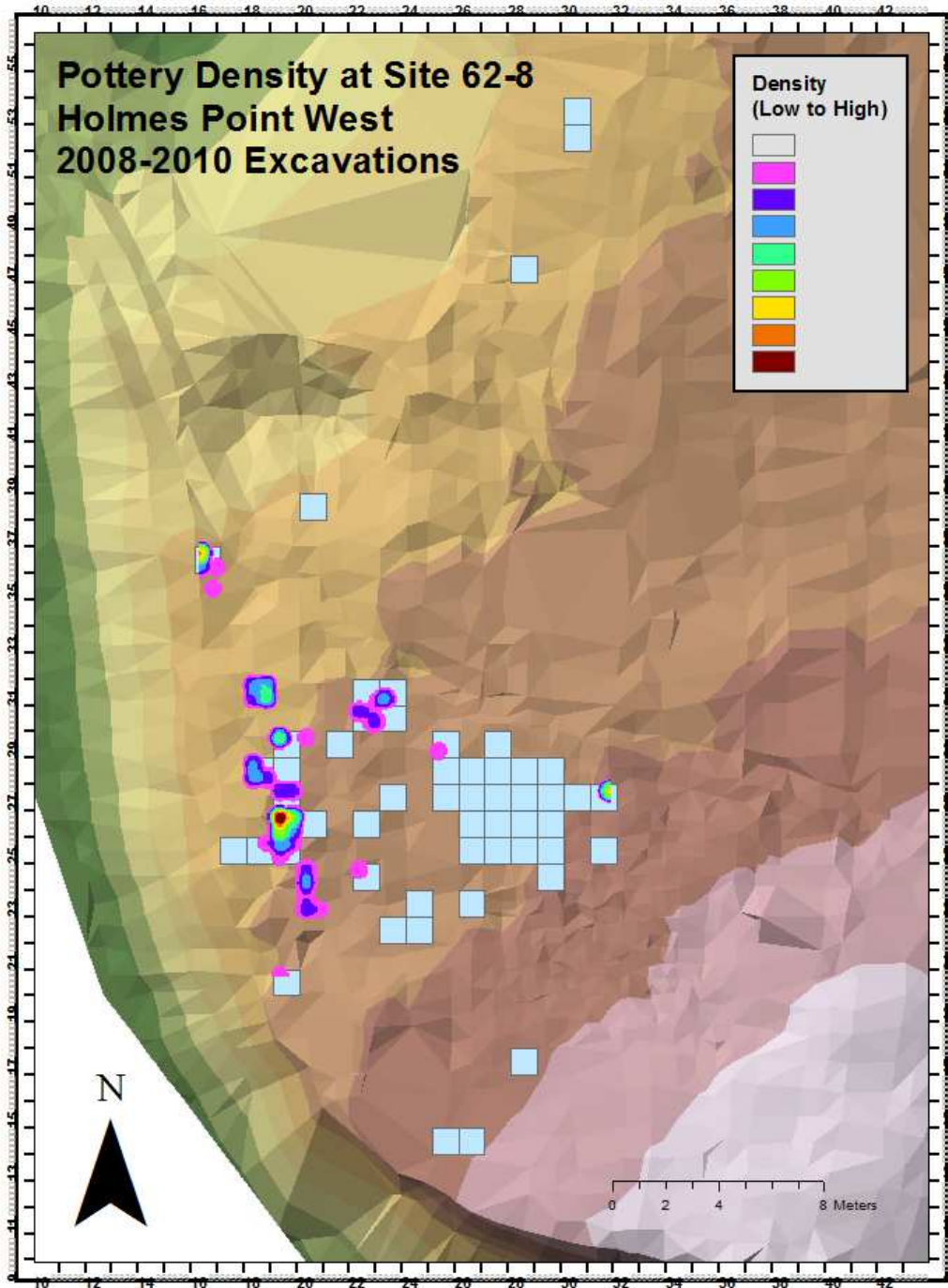


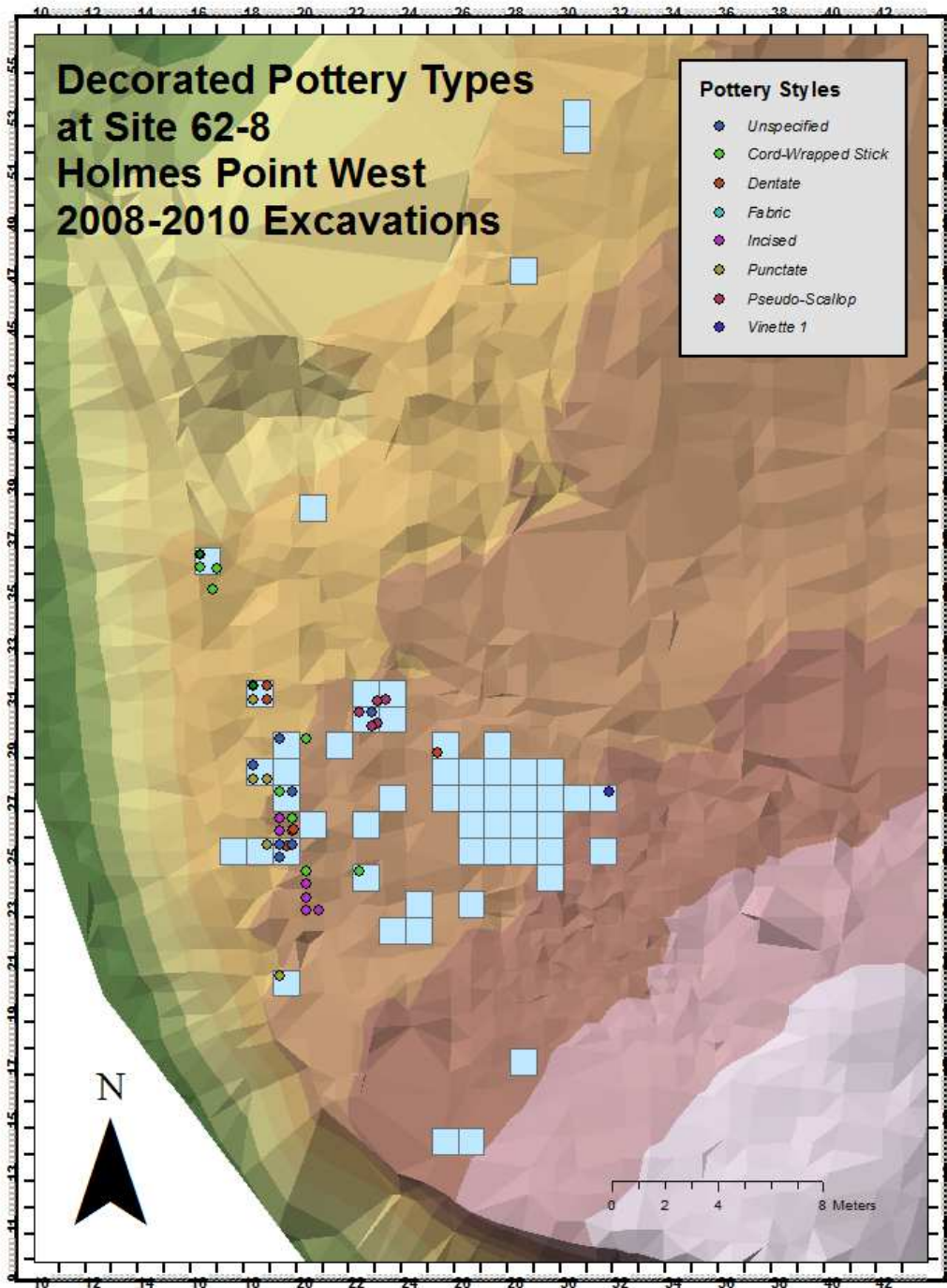




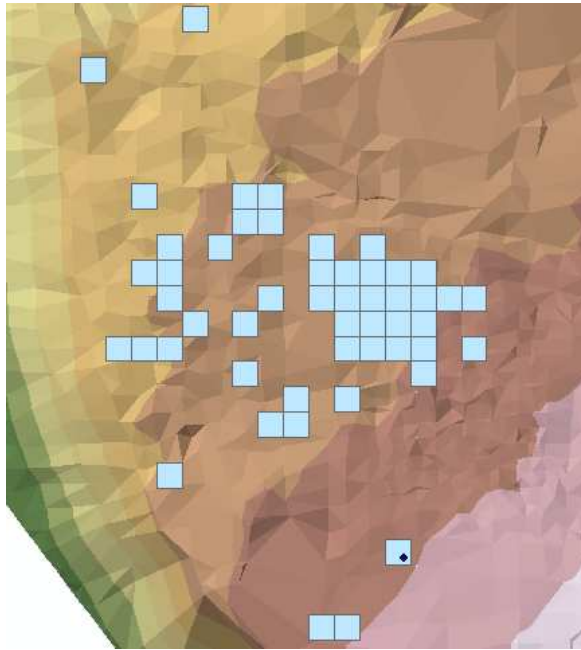




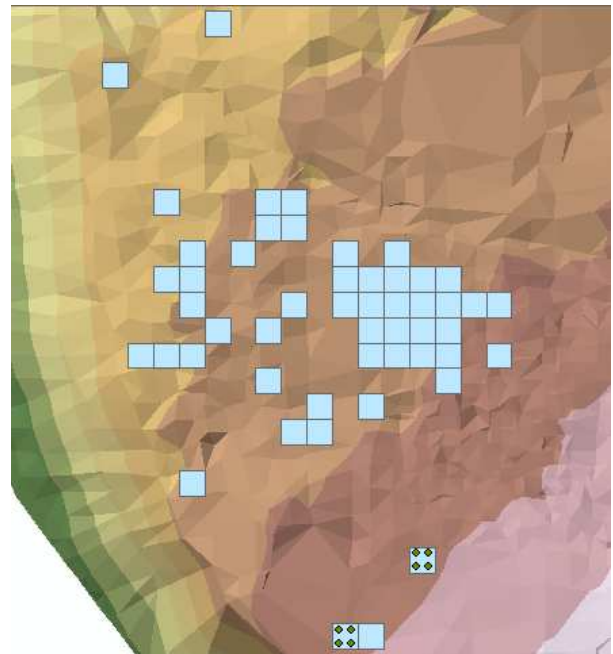




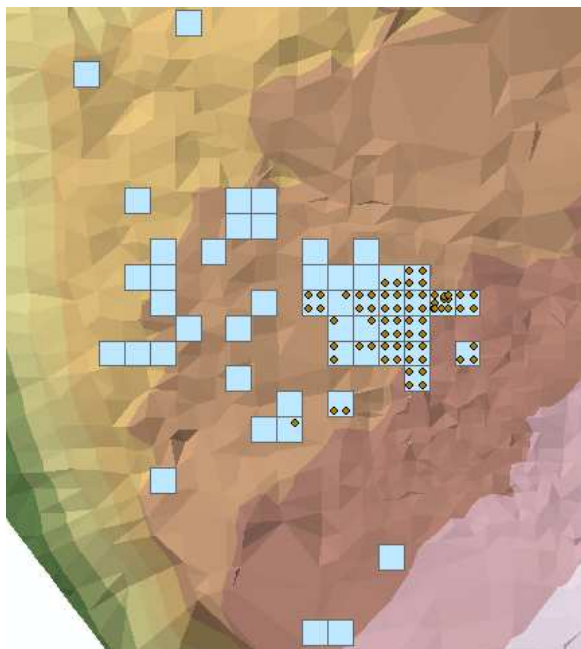
Flake Locations in the Central Area of Holmes Point West (62-8)  
By Depth Below Datum



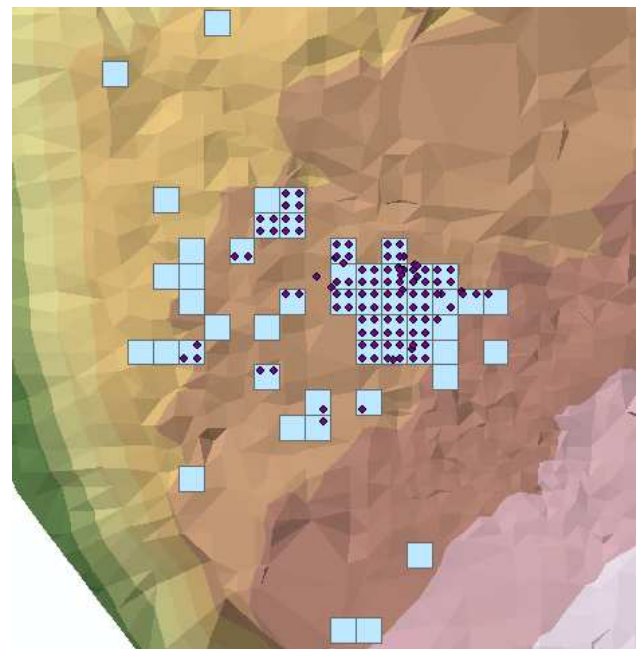
0-50 cm. BD



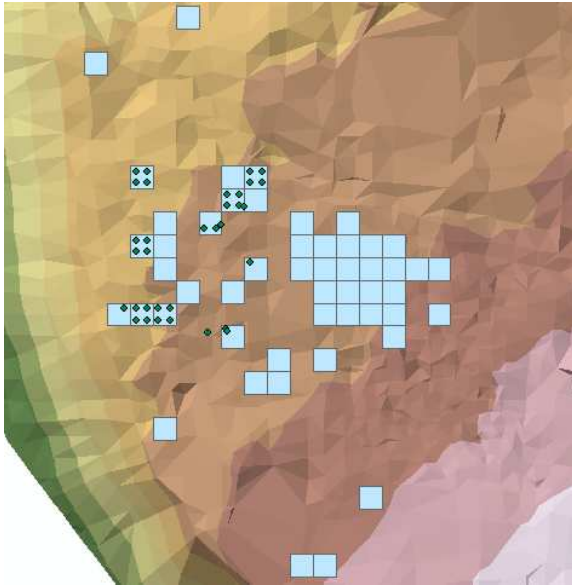
51-100 cm. BD



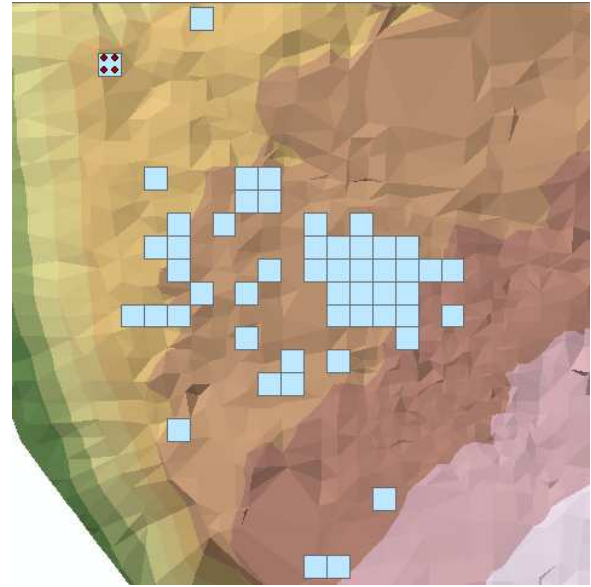
101-150 cm. BD



151-200 cm. BD

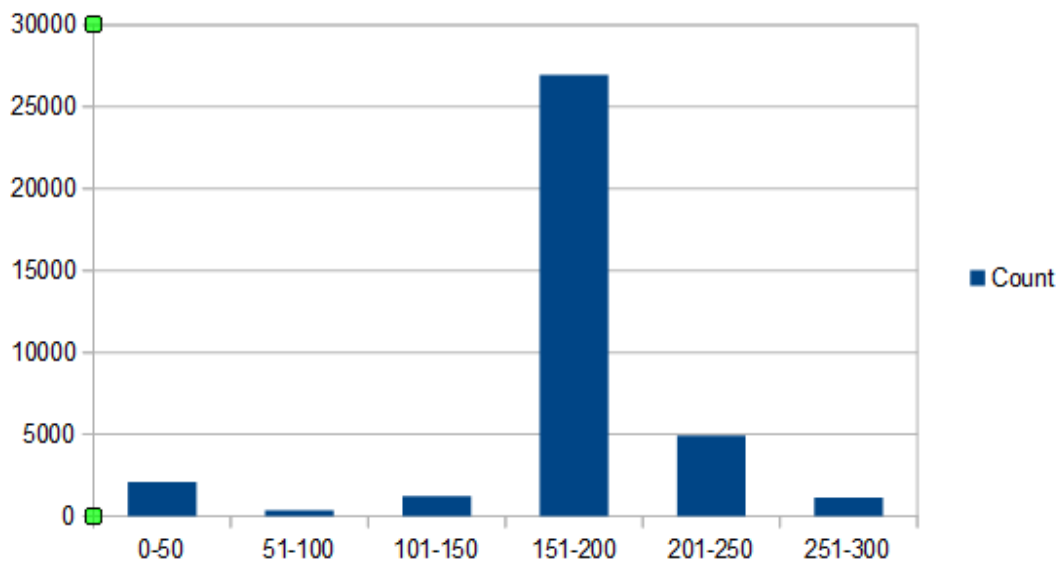


201-250 cm. BD



251-300 cm. BD

Graph of Flake Quantities by Depth Below Datum



### Appendix C: Data Dictionary

#### Fields Assigned to Holmes Point Catalog and Related Tables

ID	Unique ID assigned as entries are put into the Access database.
PN	Provenience number assigned to quad, feature, or other unit in the field.
Cat	Catalog number of artifact(s)
N	North coordinate of pit
E	East coordinate of pit
Quad	Quad within pit (NE, SE, NW, SW)
Ex_Year	Excavation year
N_Wall	Distance from north wall (for in situ artifacts)
BD_top	Top depth below datum
BD_bot	Bottom depth below datum
BS_top	Top depth below surface
BS_bot	Bottom depth below surface
Lvl	Excavation level
Strat	Stratum
Scrn	Screen size used
Soil	Context in which artifact was found
Fea	Feature number
Code_A	Top-level artifact classification code

Code_B	Secondary artifact classification code
Description	Description of artifact
Cult	Culture to which artifact belongs
NB	Supposed to signify interesting artifacts?
L_or_Less_Than_10	Length of single artifact, or number of them less than 10 mm. in size.
W_or_10_to_30	Width of single artifact, or number of them 10-30 mm. in size.
T_or_Great_30	Thickness of single artifact, or number of them greater than 30 mm. in size.
Cnt	Total number of artifacts
Weight	Weight of artifact(s)
Rock_Type	Type of rock (if lithic)
Color	Color of artifact
Disc	Has the artifact been discarded?
Ent_By	Initials of person cataloging artifact

## Fields Assigned to Pit Shapefiles

FID	Feature ID (assigned when shapefile is created)
Shape	What kind of shapefile (polygon)
ID	Not sure (all are null)

## Fields Assigned to Script-Generated Point Shapefiles (before Join)

FID	Feature ID
Shape	What kind of shapefile (Point)
ID	Not sure (null)
IDnum	Carries over ID number from original CSV file so it can be used in a join.
PN	Provenience number
Cat	Catalog number
adjusted_X	Adjusted X coordinate
adjusted_Y	Adjusted Y coordinate