



Universiteit  
Leiden  
The Netherlands

## Visualizing cityscapes of Classical antiquity : from early modern reconstruction drawings to digital 3D models

Piccoli, C.

### Citation

Piccoli, C. (2018, May 16). *Visualizing cityscapes of Classical antiquity : from early modern reconstruction drawings to digital 3D models*. Retrieved from <https://hdl.handle.net/1887/62359>

Version: Not Applicable (or Unknown)

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/62359>

**Note:** To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/62359> holds various files of this Leiden University dissertation

**Author:** Piccoli, Chiara

**Title:** Visualizing cityscapes of Classical antiquity: from early modern reconstruction drawings to digital 3D models

**Date:** 2018-05-16

## 6. Enhancing Koroneia's GIS survey data with the third dimension: A procedural modelling approach

*'The Virtual-models are there to be worked with and on rather than consumed, and can and must be negotiated, modified and engaged in the exploration of connections and context inherent in the twin processes of collage and montage that Shanks sees as central to a developing mode of archaeological practice.'* (Gillings 1999, 253).

### 6.1 Introduction

This chapter presents the methodology that I have adopted to create a GIS-based 3D visualization of the ancient town of Koroneia. The implementation of an intellectually transparent 3D visualization is particularly important in this study since comparative materials from other sites have been used (see previous chapter), and educated guesses play a major role in the reconstruction process. As discussed in chapter 4, there are, in fact, still many unknowns in Koroneia's urban layout both because the processing of the data collected during fieldwork is still in progress, and because some areas of the town are not accessible for survey due to overgrown vegetation and property's restrictions.

As highlighted in chapter 3, which reviewed current approaches to the use of 3D modelling in archaeology, the creation of intellectually transparent 3D models has been the target of only a handful of researchers, despite recent initiatives related to the issuing of guidelines on best practices and the launch of field specific journals that address these topics.<sup>1217</sup> Even more underrepresented are the applications focusing on the creation of a research-driven 3D visualization acting also as a heuristic and analytical tool. At Koroneia, the purpose of the 3D visualization was to enhance our understanding of the survey data and to use it as a tool to visualize and test different reconstruction hypotheses of the town layout. We assume in fact that the morphology of the terrain must have greatly influenced urban planning, posing constraints and offering opportunities for the laying out of the townscape. The adopted methodology required therefore to include the terrain morphology in the model in order to reconstruct the built environment – which heavily depends on the slope of the terrain. Such visualization would make it possible to investigate how buildings were constructed over the sloping ground and how many would fit within the estimated city wall circuit to formulate hypotheses regarding the town's population size.

The workflow to reach this result had therefore to be segmented across several software packages, that needed however to guarantee an optimal level of interoperability to avoid data loss in the export and import phases. Specifically, the 3D modelling software had to be able to handle the survey data that are stored in GIS file formats, to create a scene that could be exported in a format compatible with a game engine for the creation of the interactive navigation, and at the same time to be suitable for the efficient modelling and updating of a large built environment. In choosing the most suitable software packages, preference was given to a software that is supported by a large community and that for this reason has the highest probability of being maintained and preserved. At an early stage of this research, Autodesk 3D Studio Max (3DS) was used to create the first conceptual 3D models of Koroneia, in which different possible layouts of the ancient city were tested based on the preliminary results of pottery and architectural survey data combined. The choice fell on 3DS as this is one of the most used Computer Graphics (CG) software together with others such as Maya, Cinema 4D and the open source

---

<sup>1217</sup> See chapter 3, § 3.4.

Blender and it offers a variety of modelling tools and techniques that satisfy the needs of a wide range of applications for 3D models.

In our case, however, this manual modelling approach proved to be not ideal for several reasons. First of all, CG software generally do not support the handling of GIS based data (i.e. the correct retention of their spatial reference system),<sup>1218</sup> which was instead an important requirement for the Koroneia case study, both as we needed to replicate the import of GIS data several times and because we wanted to maintain a visual relationship between original data and 3D reconstruction. Secondly, the 3D visualisation of an entire city – and not just of a single building – poses additional challenges. Besides an efficient modelling of the environment in which a good balance is struck between polygons count and desirable level of detail, there is also the need to create realistic looking variations in the urban fabric. To this end, in 3DS I initially created a set of houses slightly different from one another in regard to the building arrangement and the textures used. For texturing, I made two different images for the houses' facades and four different images for the roofs and I assigned them to the houses. I then combined these slightly different houses in an *insula*-like arrangement, and manually positioned them on the terrain matching the available survey results.

This process was however quite time consuming and it did not provide satisfactory results in terms of realistic variations. Moreover, it would have been very inefficient on the long term, as a great amount of manual intervention is needed to change the 3D visualization in a CG software when new archaeological data become available, especially if considerable modifications are expected over the years as in the case of Koroneia where the study of the material is still in progress. Besides becoming clear that a more automatic approach was best suited for our case study, another requirement for our project was to work

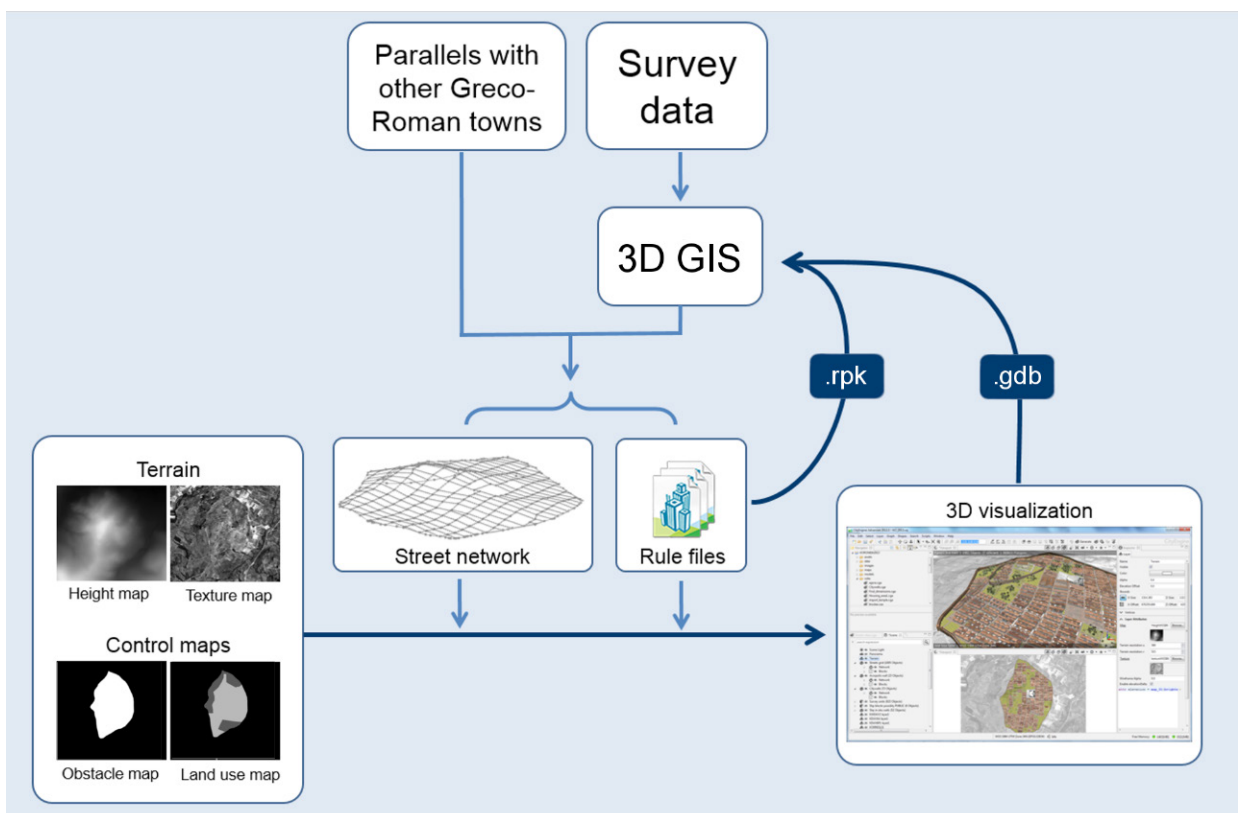


Figure 6.1 The workflow for the creation of Koroneia's 3D visualization.

<sup>1218</sup> In Blender, the interoperability with GIS has been addressed by the add-on BlenderGIS which enables users to import and, with some limitations, export geospatial data (<https://github.com/domlysz/BlenderGIS>).

in an environment that we could use more as a 'laboratory' to formulate hypotheses instead of only as a visualization of existing knowledge. While CG software packages are suitable for simulations, such as on the use of light and the perception of space as previously discussed, they were less appropriate in our case, where we wanted to combine several layers of information that could function as a tool to enable a more analytical approach towards the built environment.

To fulfil these requirements, we employed advanced techniques based on GIS and procedural modelling that are employed in geo-design and modern urban planning, and that allow a high degree of automation in the process of transforming 2D data into a 3D environment. This chapter presents the workflow that I have developed, which consists of three phases: 1) 3D mapping of the finds onto the Digital Elevation Model of the hill to visualize their distribution and characteristics as recorded during fieldwork; 2) reconstruction of the ancient terrain and the possible layout of the *polis* in the 4th century BC, which corresponds to a moment of possible disruption in the urban life resulting from the *andrapodismos* that the Koroneians suffered in 346 BC;<sup>1219</sup> 3) experimentation with setting up an interactive first-person walk-through to allow a more experiential navigation in the virtual environment and with implementing a platform to communicate our interpretation in the future. Additional elements will be discussed, such as the contribution of this methodology towards the implementation of intellectually transparent and reusable 3D reconstructions. The chapter closes with a discussion on the advantages and disadvantages of using the suggested approach and some possible future applications for Koroneia's 3D environment.<sup>1220</sup>

## 6.2 Workflow

The approach that I have adopted to deal with the data of this complex urban site is based on the exploitation of GIS in combination with procedural modelling techniques. CityEngine, whose characteristics have been already presented in chapter 3, has been chosen as the procedural modelling software because of its modelling capabilities and its interoperability with ArcGIS. While CityEngine has been already used for the reconstruction of several archaeological sites,<sup>1221</sup> the methodology here presented exploits to the fullest the connection between CityEngine and ArcGIS in order to create a 3D GIS that comprises both the original survey data and the reconstruction hypotheses of the ancient urban layout.

The adopted workflow is summarized in figure 6.1. Comparisons with other Graeco-Roman towns in the Boeotia region and elsewhere in Greece created the framework for the interpretation of the survey data of the Boeotia project, many of which were already stored in a GIS environment.<sup>1222</sup> As will be described in more detail in the following paragraphs, I have written *ad hoc* procedural rules that I have applied 1) as a procedural symbology layer in ArcGIS, which exposes the properties of the architectural dataset (i.e. dimensions and stone type) in an automatic and more customizable way than standard 3D symbology; and 2) to the reconstructed terrain and street network to create a 3D reconstruction hypothesis of the ancient *polis*.

Importantly, each of the components that is included in this methodology and in the chart displayed in figure 6.1 can be modified in isolation from the others, thus enabling an efficient workflow given that data are still being collected in our study. Moreover, the same rule can be applied to an updated dataset, making it possible to iteratively change the 3D scene when new data are available or new hypotheses have to be visualized.

<sup>1219</sup> See chapter 4, p. 92

<sup>1220</sup> The ongoing results of this project have been published in Piccoli 2014 and Piccoli 2016.

<sup>1221</sup> See chapter 3, p. 58.

<sup>1222</sup> By Bart Noordervliet and Janneke van Zwielen.

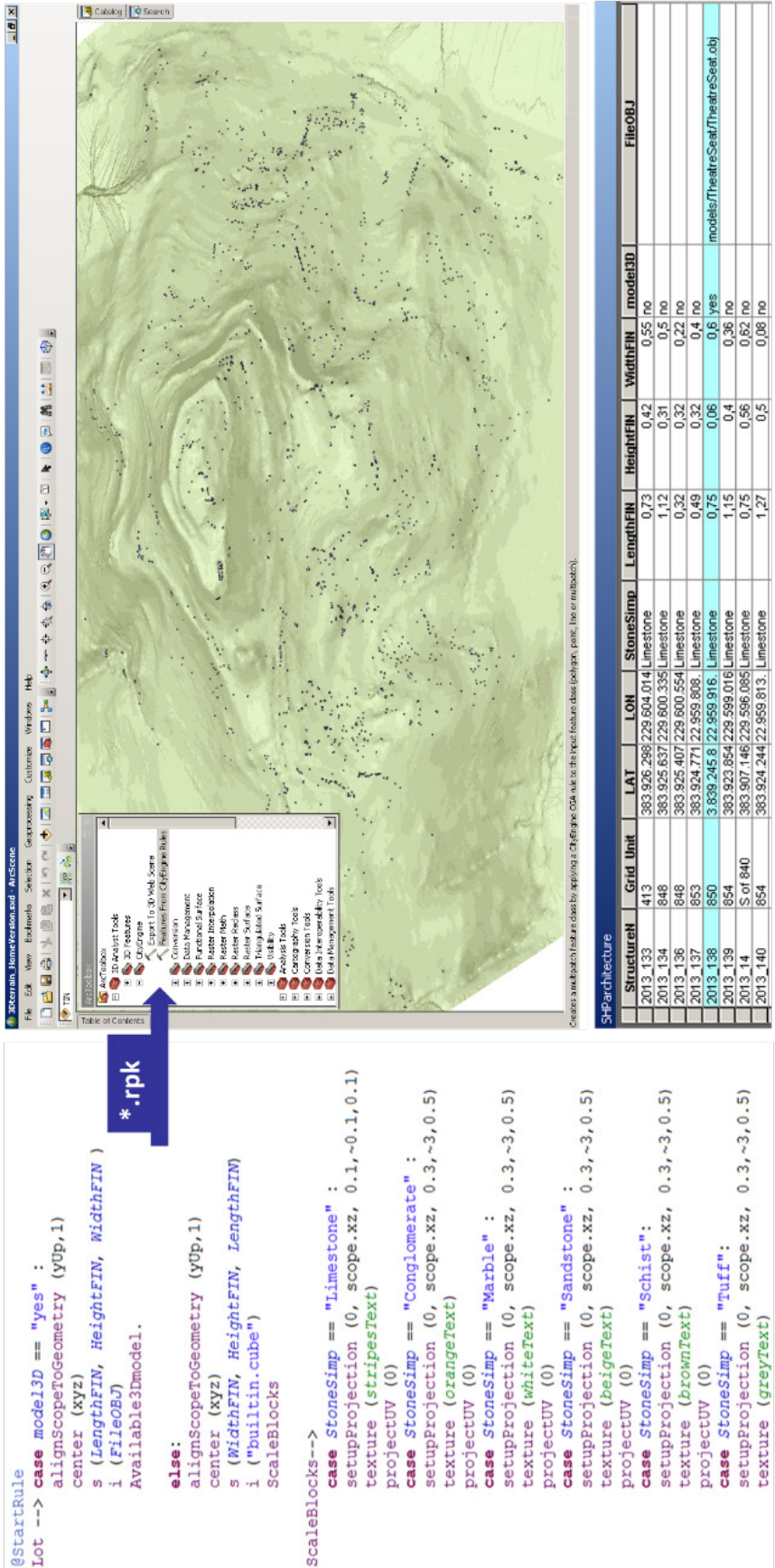


Figure 6.2 Left: Part of the procedural rule written in CityEngine which evaluates the information contained in the specified columns in the architectural pieces' shapefile (e.g. LengthFIN, HeightFIN and WidthFIN) to modulate size and appearance of the architectural finds based on their characteristics as recorded in the field.

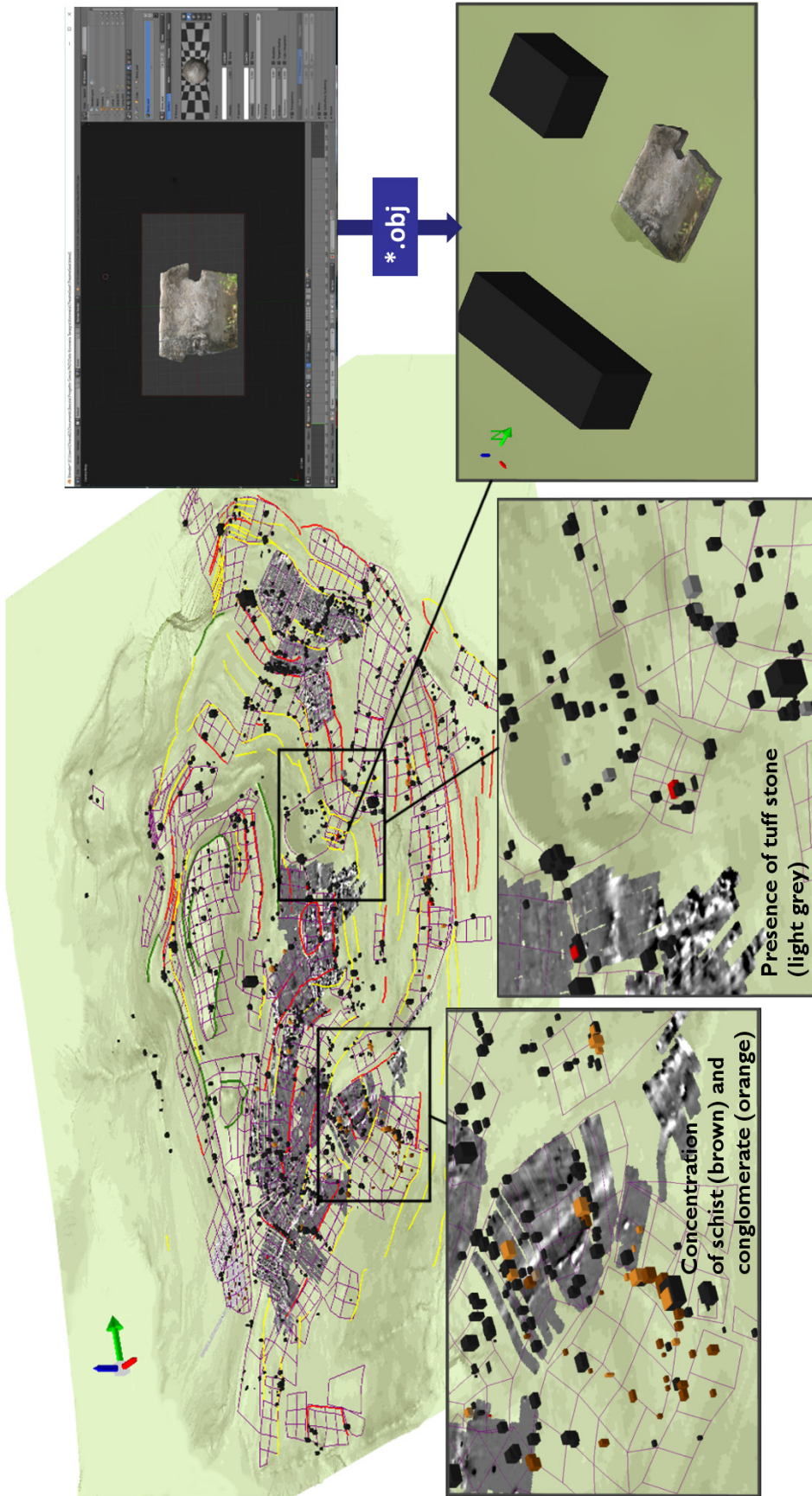


Figure 6.3 An example of some data layers of Koroneia's 3D GIS: the DEM with overlaid grid of the areas that were accessible during the survey, the terraces mapped during the geomorphological survey, the provisional results of the ongoing geophysical prospections, and the architectural finds. The latter are scaled according to their dimensions (here multiplied by a factor of 10 for visualization purposes) as recorded in the field, and categorized per stone type with a procedural rule that was written in CityEngine and imported in ArcGIS. In this way, clusters of stones, the relationship between the dimension of the blocks and the stone type, and special finds (of which 3D models are automatically imported as OBJ files) become immediately apparent.

### 6.2.1 A rule-based 3D GIS of architectural survey data

The aim of the first phase of this work was the creation of a 3D GIS of the survey data that are currently available, which enables us to have an intuitive overview of the relationships between various data layers, such as terrain morphology, terraces and concentration of finds. To this end, I wrote an *ad hoc* procedural rule in CityEngine (see Figure 6.2), which interrogates the attribute table of the shapefile containing the architectural elements, substitutes the latter with blocks that are scaled according to the real dimensions of the finds as recorded in the field (i.e. length, width and height), and associates different textures to the blocks based on the types of stone they are made of. The rule is exported from CityEngine in \*.rpk (rule package) format, which compresses the rule and related assets (e.g. textures), and is applied to the selected 2D features using the CityEngine toolbox that has been introduced in ArcGIS 10.2 among the 3D Analyst tools (see Figure 6.2).

Specifically, the shapefile of the architectural pieces contains information on their dimensions (in columns LengthFIN, HeightFIN, and WidthFIN) and on their stone type (column StoneType). Moreover, I have added an additional column (FileOBJ) that contains the relative path to a more detailed (manually modelled) 3D model of the architectural piece in case this is available (see Figure 6.2, bottom). Besides modulating size and appearance of the architectural blocks, the rule includes the possibility to automatically substitute 3D models of special finds to their corresponding position on the DEM of the hill. For this purpose, I modified the attribute table of the shapefile containing the architectural blocks' coordinates and description by adding two columns, one storing the Boolean values 'yes' or 'no', which indicates the availability of a 3D model, and the other containing the relative path of the 3D model's file (see Figure 6.2). The pictures that were available of the finds were not suitable for a photogrammetry-based 3D model; therefore, I manually modelled some of the architectural finds in Blender using the pictures and the measurements taken on site as reference. This is the case for example of the carved limestone block (catalogue nr. 2013\_138) that was found in the vicinity of the location of Koroneia's theatre, now lost, and initially identified as a possible theatre seat. The 3D model of the piece was then saved in OBJ format and automatically imported in ArcScene with the above mentioned rule file.

This workflow therefore extends the current possibilities offered by ArcGIS 3D symbology, which allows the user only to automatically stretch a flat 2D shape according to a predefined height (i.e. to vertically extrude lines and polygons) and to categorize the finds with different textures. The procedural rule, instead, allows the automatic scaling of the initial 2D shape according to all three dimensions (width, depth and height), and the automatic import of pre-made 3D models of finds on their correct location. This results in a more realistic visualization of the surveyed pieces, and thus in a more intuitive interpretation of the finds.

#### *Interpretative visualization for intuitive insights into data clusters*

By applying in ArcScene the procedural rule that interrogates the attribute table of the shapefile containing the architectural elements, and modulates their size and appearance according to their dimension as recorded in the field, concentrations of smaller or bigger blocks, clusters of specific types of stone or special finds become immediately visible. In addition, possible correlations between the dimensions of blocks and their stone type can be easily spotted by zooming into the 3D scene (Figure 6.3). Among the visible clusters that clearly stand out after applying the above mentioned rule, there is a concentration of schist and large conglomerate blocks, which emerges from the predominant limestone (textured in black) to the south-east, close to the city wall circuit. Assuming that larger blocks were used for more impressive architecture, such as buildings with a public character, this concentration may point towards the presence of a relatively large building, perhaps a church or



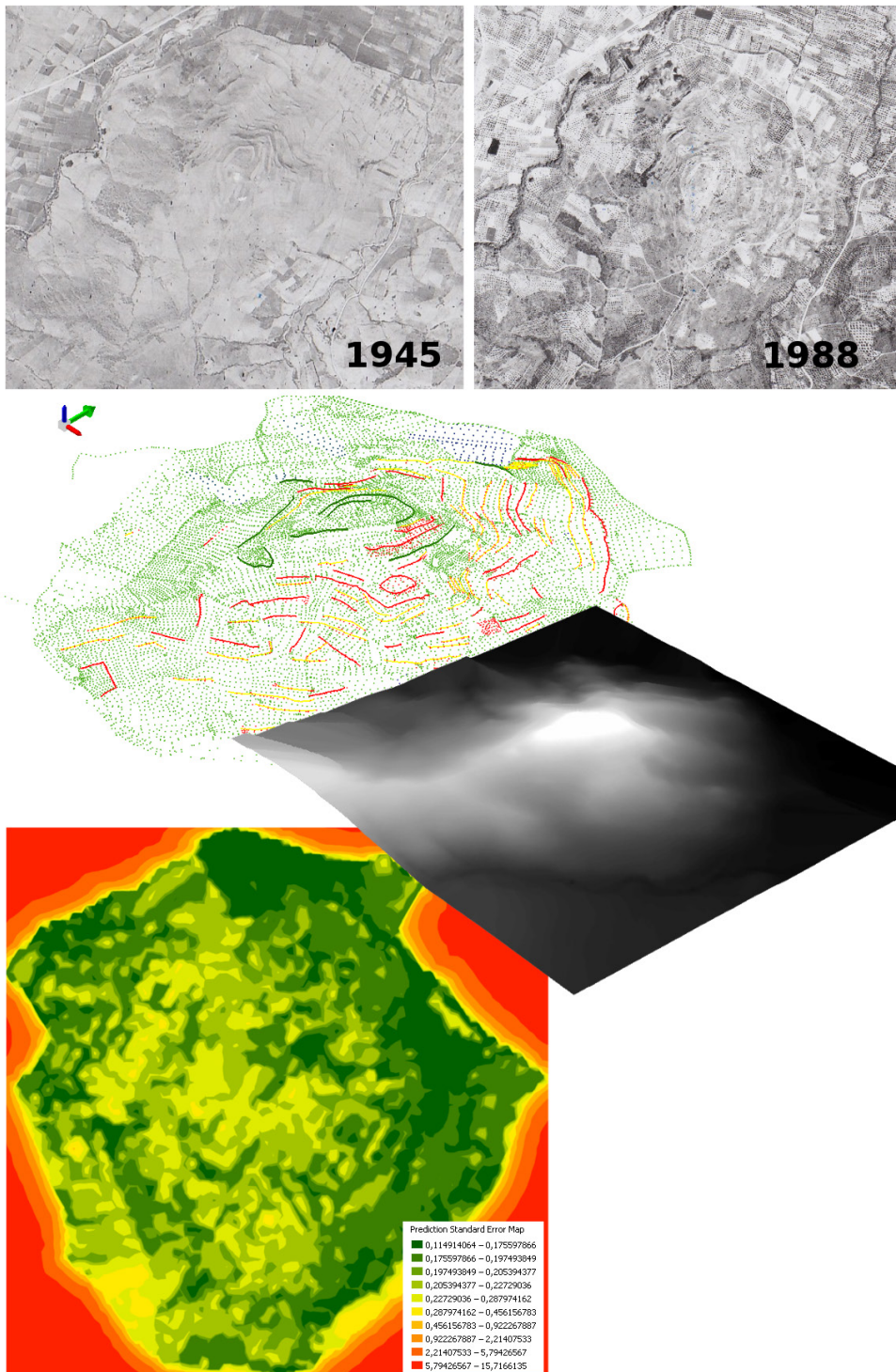


Figure 6.4 Top: Examples of historical aerial imagery that was used together with the terraces mapped by age (overlaid on the DGPS points representing the current terrain morphology) as guides to create a reconstruction hypothesis of the terrain in antiquity. Green lines represent ancient terraces, red lines modern terraces, yellow lines terraces of uncertain age. Points were added to (represented in blue) or subtracted from those originally recorded on the hill during the microtopographic survey. As a result, a new DEM was created by interpolating the points in ArcGIS using a Kriging interpolation method (see prediction error map). This DEM was then stylized as a grey scale image and exported in TIFF format to be used as a terrain heightmap in CityEngine.

a villa.<sup>1223</sup> Moreover, a high number of blocks made of tuff, a relatively soft stone that lends itself to carving and moulding, is visible on the plateau where the agora was located and on the nearby hollow which was occupied by the theatre.

This rule allows, moreover, the automatic placement of the textured 3D models of architectural finds directly on their findspot. This solution is particularly useful for including 3D models of special architectural elements, such as decorated or uncommon pieces, that can therefore be easily spotted by navigating the DEM in the 3D GIS environment. As visible in Figure 6.3, the 3D model of a carved block that I have created in Blender from pictures and measurements taken *in situ* has been automatically loaded onto the position of the piece as was recorded by the GPS during fieldwork.

### 6.2.2 Reconstruction of the ancient terrain morphology and urban layout

The second phase of the workflow comprised the modelling of a reconstruction hypothesis both for the ancient terrain and for the ancient town. The main sources for the creation of the ancient terrain were the results of the geomorphological survey conducted by Keith Wilkinson<sup>1224</sup> and historical aerial imagery which I have used as a comparison to trace back the modifications that the hill underwent. Specifically, images from 1945, 1960 and 1988 acquired by the project from the Ministry of Agriculture show the progressive agricultural intake of the hill by farmers and the creation of paths to navigate around the hill (see Figure 6.4). The points that were recorded in the field with a DGPS were deleted and new ones were added in ArcScene where modern modifications needed to be smoothed off or filled. In order to facilitate the interpolation of the new DEM, for example, I added points to fill in the large quarry that was opened on the north-western side of the hill in modern times and the two gullies that were caused by rainwater washing away the slopes made of unstable low grade geology (Figure 6.4).

It must be noted that the goal of this process was to reach a plausible hypothesis of the shape of the terrain before the beginning of the agricultural work, one of the main causes of change in the hill morphology. While at this stage it is impossible to ‘reconstruct’ with certainty the terrain morphology as it used to be in antiquity, it is however possible that the continuation of the archaeological investigation and the geomorphological survey will offer additional data to reconstruct the level of the ancient terrain at least in some parts of the town, e.g. based on the depth of the buried archaeological finds in case of excavation. For this reason, I



Figure 6.5 A TIFF image of the geophysical results and the *in situ* walls (in red) is imported as map layer in CityEngine, thus allowing the retention of the visual and spatial relationship between the original data and the reconstructed street network.

<sup>1223</sup> See chapter 4, p. 126.

<sup>1224</sup> See chapter 4, p. 106.

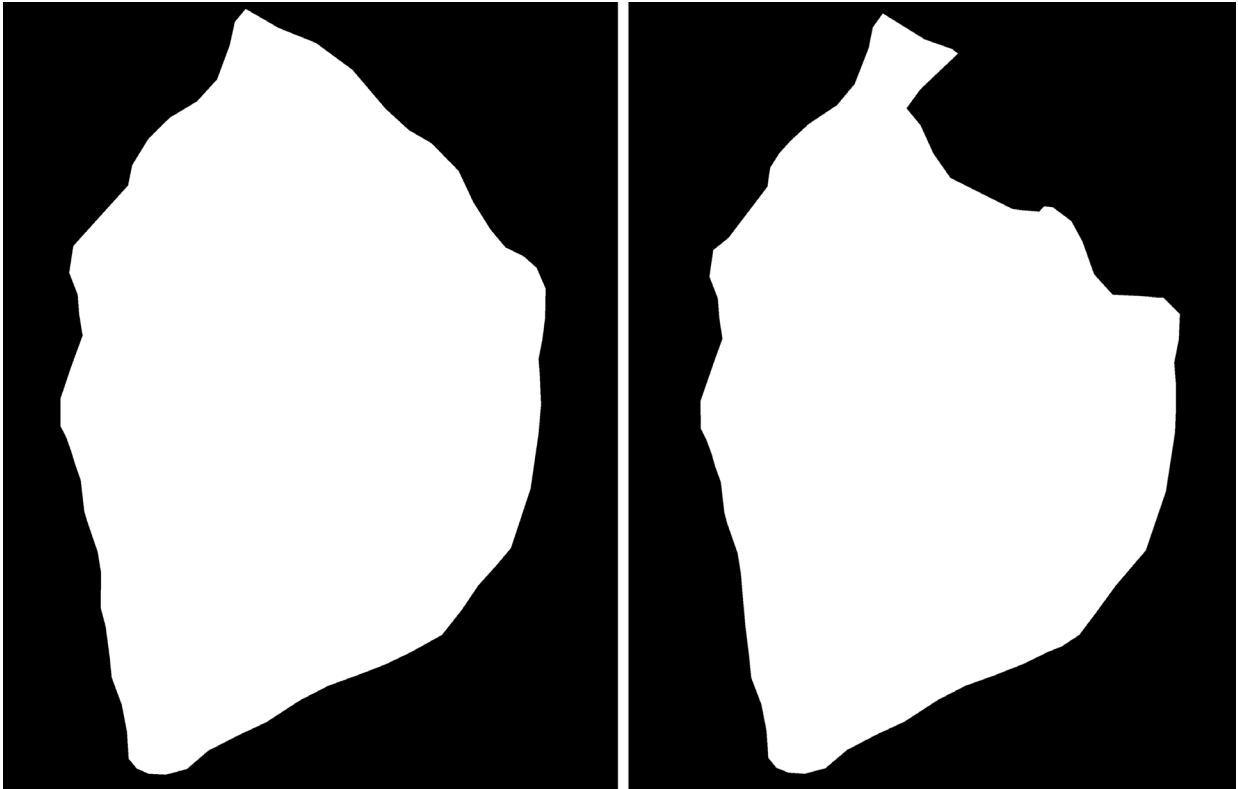


Figure 6.6 The two obstacle maps that limit the generation of the 3D environment only within the white area. Left: the obstacle map covers the largest town extent with the northern stretch at the foot of the slope according to the surveyed architectural remains; right: obstacle map covering the town extent when the geophysical features are considered (for the discussion of the available archaeological evidence, see chapter 4).

have documented all the steps I took in creating the new DEM. Specifically, I have stored all the points that I have deleted or added in separate shapefiles in order to keep the record of the operations that I have performed to change the DEM. After some experimentation and comparison between models created with different interpolation methods, a terrain model resulting from a Kriging interpolation method was used.<sup>1225</sup> The output raster was subsequently stylized as a grey scale image (stretch type: Minimum-Maximum) and exported in TIFF format to be used as a height map for the terrain generation in CityEngine (Figure 6.4).

#### Map layers

CityEngine offers several so called map layers for the customization of the 3D environment; specifically these are the terrain, texture, obstacle, mapping, and function layers.<sup>1226</sup> The use of map layers is very advantageous to control the city generation, especially in our case when new data can modify our

<sup>1225</sup> Ordinary Kriging, model type: Gaussian; Prediction Errors (samples 18547 of 18547):

Mean: -0,013356726893272162

Root-Mean-Square: 0,4033660270359244

Mean Standardized: -0,020829271931495624

Root-Mean-Square Standardized: 0,7823976642156636

Average Standard Error: 0,5374845926168181. The data evaluation and models comparison were carried out following these resources: <https://www.azavea.com/blog/2016/09/12/kriging-spatial-interpolation-desktop-gis/>; <http://desktop.arcgis.com/en/arcmap/latest/extensions/geostatistical-analyst/comparing-models.htm> (accessed Sept. 2016).

<sup>1226</sup> For an overview of map layers and their usage, see the CityEngine user manual at [http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/al/edit\\_al\\_f.html](http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/al/edit_al_f.html) (accessed Sept. 2016).

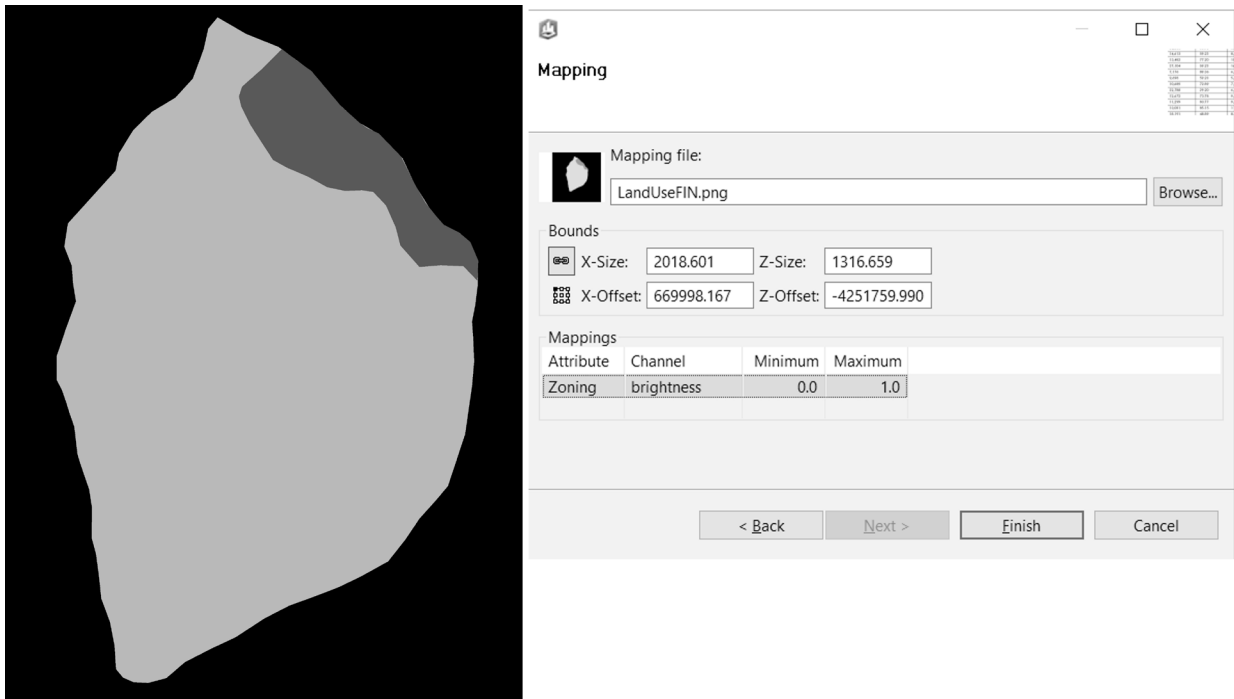


Figure 6.7 Left: TIFF image loaded as attribute layer in CityEngine to guide the creation of different land uses; Right: Window menu that allows the creation of an attribute (i.e. "Zoning") associated with the land use map.

```

import rule : "Individual_houses_slope.cga"

attr Zoning = 0
attr red    = "#FF0000"
attr green  = "#00FF00"
attr blue   = "#0000FF"
attr white  = "#FFFFFF"

@StartRule
LandUseType =      case Zoning > 0.5 && Zoning < 1 :
                    "buildingLot"
                    else : "vegetation"

Lot --> case LandUseType == "buildingLot" :
        rule.Lot
        case LandUseType == "vegetation" :
        report ("Area.Greenspace", geometry.area) rule.LotVegetation
        else : NIL

Garden --> report ("Area.Greenspace", geometry.area) rule.Garden
    
```

Figure 6.8 Rule file that guides the mapping of different land uses according to the input image represented in Figure 6.7.

knowledge about the ancient morphology or the urban layout. Each of the images that represent a layer can in fact be updated and the visualization will adapt accordingly to the new input data. By using map layers, it is moreover possible to easily generate different reconstruction hypotheses in a time efficient manner. To my knowledge, this possibility has never been exploited before in other archaeological projects that have used CityEngine as their main software environment.

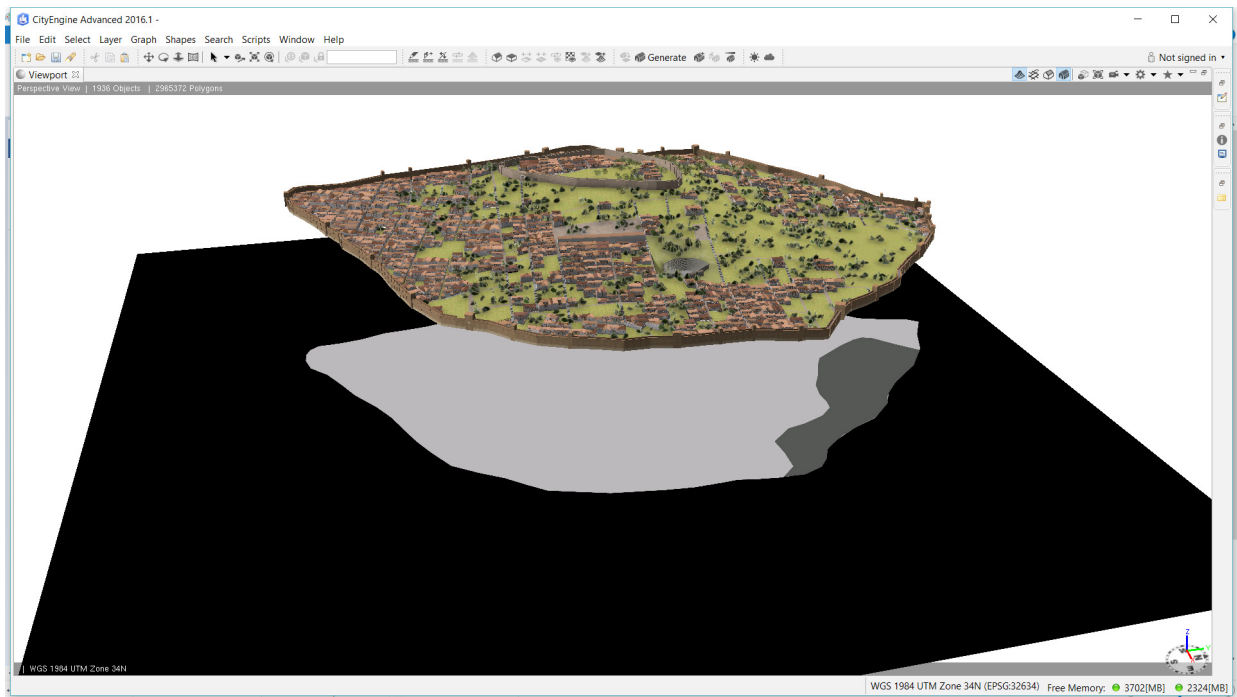


Figure 6.9 Screenshot from the CityEngine viewport showing the results of the application of the attribute map (visualized as the black and grey image underneath the reconstructed town). This map assigns the rule for the creation of buildings to the light grey area and the rule for the generation of vegetation to the darker grey area. In this way, it is possible to quickly create two reconstruction hypotheses that show the northern area of the town as either sparsely or more densely built up.

The terrain layer is created by importing a heightmap and a texture and by defining the minimum and maximum elevation values to which the greyscale image corresponds. Since the heightmap for Koroneia was created by modifying the georeferenced DEM of the hill, the TIFF file already contained the correct spatial reference which was then automatically recognized by CityEngine upon import. Additionally, I have created a TIFF image containing the geophysical results and the *in situ* walls in ArcGIS and imported this as an additional map layer. In this way, I was able to keep the reference with the geophysical data while creating the reconstruction hypotheses for the urban layout (Figure 6.5).

Another map layer that I used for Koroneia's case study is the 'obstacle map', a black and white image that defines a Boolean attribute used to limit the generation of the 3D scene within its white areas. I have created such images in GIS by drawing a polygon over the supposed perimeter of the city wall (see chapter 4, Figure 4.32), which was assigned a white colour, and by stylizing the underlying DEM with a black colour (see Figure 6.6). Since parts of Koroneia's perimeter are uncertain, I created two separate obstacle maps. One obstacle map represents the largest projected circuit on the northern slope (the orange line in Figure 4.32) which projects the line of the city wall stretch identified during the architectural survey at the north-western foot of the hill. The other obstacle map stops instead at the features identified by the geophysical survey on the northern slope as a possible city wall stretch and reconnects with the line of the city wall identified during the architectural survey (Figure 6.6, right following the yellow line in Figure 4.32).

Finally, I have used the so called 'mapping attribute layer' to distinguish between buildings and open spaces in the creation of Koroneia's 3D scene. In general, this layer makes it possible to assign different rule files to a lot, depending on its location. It is possible to use a RGB image or a greyscale image as a mapping attribute layer. The different colours or grey intensities are then used in CityEngine to map different land uses. In Koroneia's case, I have created a greyscale image in ArcMap to map

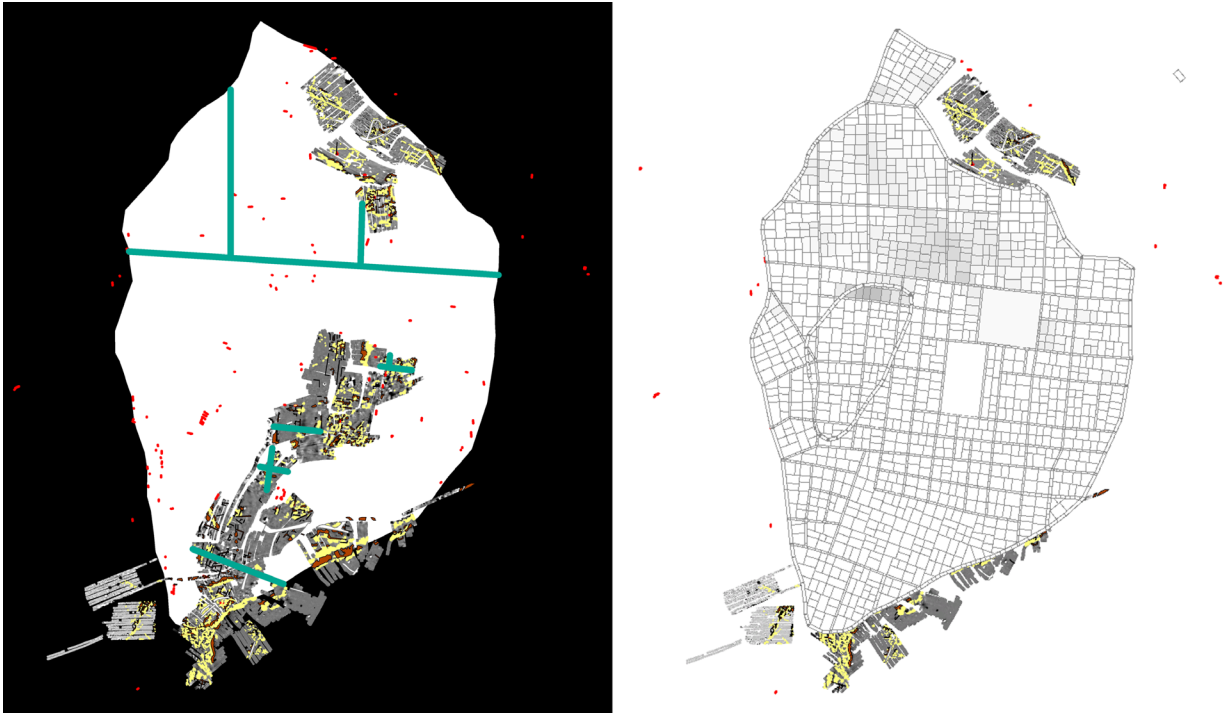


Figure 6.10 Left: Screenshot from ArcMap's workspace displaying the geophysical results and the *in situ* wall stretches which have informed the reconstruction of the street network (in blue, the lines that were stored in a shapefile and imported as input data into CityEngine); Right: The reconstructed urban grid adapted to the terrain in one of the proposed hypotheses.

two hypotheses regarding the northern slope of the town by modifying one of the obstacle maps. In more detail, I assigned a light grey shade to the intra-mural area and manually mapped the area that appears to be empty of buildings in the northern part of the town with a darker shade of grey (Figure 6.7, left). Upon import of the map layer into CityEngine, I have set an attribute ('Zoning') and specified what kind of channel had to be used to interpret the image (in this case, brightness) and the minimum and maximum values within the channel (from 0.0 to 1.0) (see Figure 6.7, right). I have then written a rule file which relates a land use to a specific range of values (Figure 6.8). Specifically, the rule containing the formal description of domestic architecture and the agora was assigned to the areas mapped with the lighter shade of grey, while the rule that creates green spaces was associated with the areas mapped in darker grey. Upon generation of the environment, the rule evaluates the parameters in the image and creates the geometry which corresponds to the assigned values (see Figure 6.9).

For the moment, this map facilitated the creation of a reconstruction hypothesis of the larger town extent showing sparse habitation on the lower northern slope. This scenario visualizes the results of the geophysical survey which has yielded only faint traces of occupation on the northern sector of the town. In the future, however, when the results of the pottery data will be available, this mapping will be useful to further refine the creation of different areas within the town – e.g. discerning putative residential and public areas – and map the expansion and contraction of the town through the centuries.

#### *Street network*

After importing the obstacle maps, I have created a reconstruction hypothesis of the street network based on the available survey results. I have used the streets that were identified during the geophysical

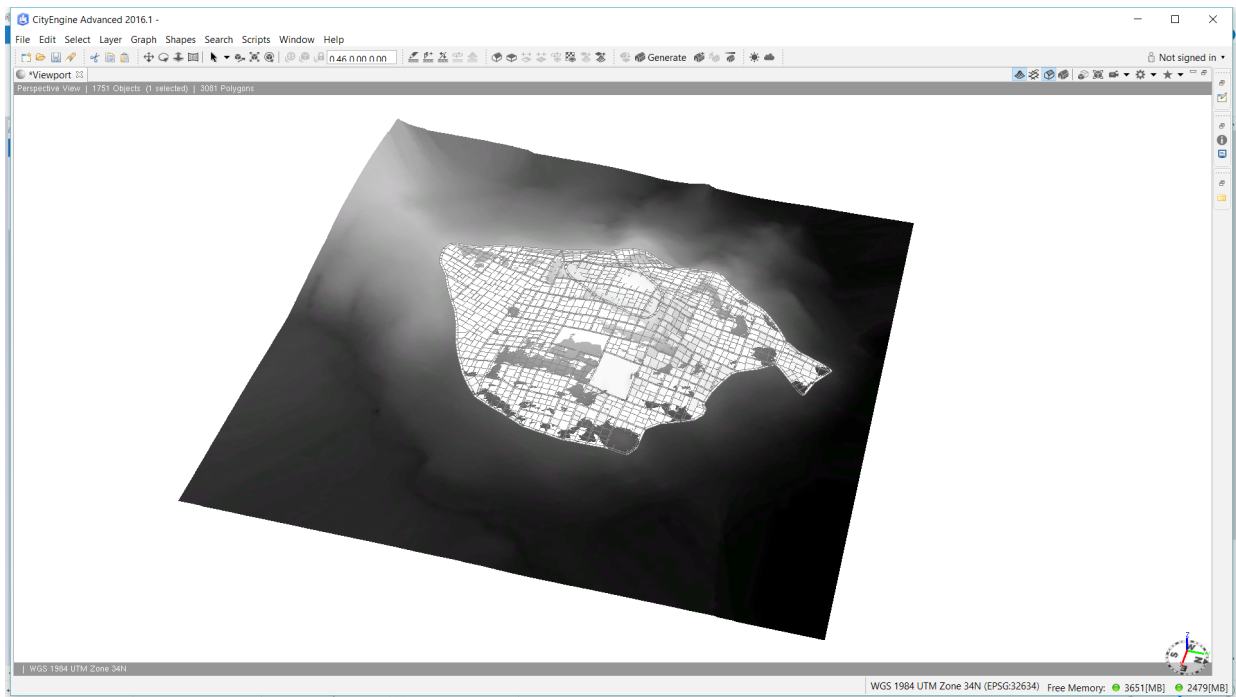


Figure 6.11 Screenshot from CityEngine showing the overlap between the terrain heightmap and the layer with streets and blocks.

survey as the starting point for the generation of the entire street system. Our working hypothesis is that the regular grid that appears both in the geophysical prospections and in the surveyed *in situ* wall lines is part of the same network and can be dated to the 4th century BC. As discussed in chapters 4 and 5, this assumption is based on comparisons with other sites where a (roughly) orthogonal grid started to be used in this century and was maintained until Late Antiquity.<sup>1227</sup> I have digitized the few streets that have been recognised in the geophysical survey in GIS, added and extended the orientation of the surveyed *in situ* wall stretches, imported into CityEngine the shapefile containing the graph and used this as input for street generation (Figure 6.10). To avoid having to manually set the streets' width, I have stored in the attribute table the width of the known streets as these values are recognized by CityEngine and are automatically used to model the correct width. I have then selected these few streets and launched the 'grow street' operation, which offers several parameters to guide the street creation. These include the possibility to set the number of streets, to decide the pattern of major and minor streets (either organic, radial or raster), and to use the environment as a constraint. Specifically, a maximum-allowed street slope can be set and the terrain and obstacle map can be given as parameters.

Each street in CityEngine is typically composed of a series of segments (major or minor edges) connected by nodes, both having as features a number of shapes which represent the surfaces creating the proper street and the adjoining sidewalks. Since the first input consisted of only a few streets, it was necessary to repeat the operation several times and include some manually drawn streets to cover the entire urban area. By running the 'align graph to terrain' tool, the nodes of the graph network have been then automatically aligned with the heightmap representing the terrain.

After creating the graph representing the street network, the software automatically filled in the spaces between streets' segments and created so called 'lots', namely adjacent polygons that are the starting point for rule based modelling. In fact, CGA rules can only be applied to surfaces. Once again, a set of parameters governs the appearance of such lots, such as the way in which they are subdivided and their

<sup>1227</sup> See esp. pp. 141 and 155.

```

// ATTRIBUTES
@Group("LEVEL OF DETAIL")
@Range (Low = 0, High = 1)
attr LOD = 0 //controls vegetation's detail
attr HighLOD = true //controls buildings' detail

@Group ("VEGETATION")
attr TreesNumber = 2
attr TreeHeightMin = 5
attr TreeHeightMax = 10

@Group ("HOUSES DIMENSIONS")
attr RoofShedSlope = 10
attr wall = 0.6
attr DoorOpeningRange = rand (1.2, 1.6) // cf. Halos, Haagsma 2003, 43.
attr WingHeightTotal = groundfloorHeight +0.4
attr groundfloorHeight = 2.2
attr firstfloorHeight = 2

```

Figure 6.12 Example of the parameters that I have set as attributes in the rule for domestic architecture. Note the addition of explanatory comments preceded in this case by a double slash (//) to exclude them from the parsed code.

```

// CATEGORIZATION OF HOUSES' FOOTPRINT SIZE
footprintSize(geometry_area) =
/* Classical houses greatly varied in size:
A database of more than 300 hundred examples of houses in the Aegean from 800 to 300 BC
shows a progressive increase in house size (Morris 2005, 108). In Athens some of the houses
in the 5th c. BC residential block on the north foot of the Areopagus were as small as 50-
70 m2 (Nevett 1999, 86), while Houses C and D near the Great Drain were 130 and 225 m2
respectively (Nevett 1999, 88). Evidence from Athens, Attica and Leucas shows that an
average sized 5th and 4th c. BC dwelling covered around 130 m2 (Konecny 2013, Fiedler 2005,
Nevett 2005, Tsakirgis 2005).
The 'typical' 4th c. BC Olynthian house of the North Hill covered around 294 m2, while the
houses on Row A were on average 350 m2 (Ault 2005, 66).
Exceptionally larger houses are also attested as the House of the Mosaics at Eretria with
an area of 625 m2 (Bintliff 2012, 300).
*/
case geometry_area <= 50: "empty"
case geometry_area >50 && geometry_area<= 70: "VerySmall"
case geometry_area >70 && geometry_area<= 130: "Small"
case geometry_area > 130 && geometry_area <= 300: "Medium"
case geometry_area > 300 && geometry_area <= 400: "Large"
else : "VeryLarge"

```

Figure 6.13 Lines from the rule of domestic architecture which categorize the houses according to their size. Note the insertion of comments, which explain the modelling choices that are recorded in the rule files. In this case the multi-line explanation is excluded by the processed script by adding /\* and \*/ at its beginning and end (cf. other scripting languages such as C).

dimensions, how they are aligned over the terrain and whether each of them should have access to the street or not. The minimum and maximum size of the blocks can be set either manually under 'Block parameters' in the inspector, or automatically by using the 'function' map layer. I chose for the latter approach as I wanted to reduce the amount of manual work that was needed to create Koroneia's 3D environment and its following updates as much as possible. The allowed lot sizes range between 50 and 400 square meters, thus giving a representative sample of the variations in house sizes that have been observed in Classical – Hellenistic sites in Greece.<sup>1228</sup>

<sup>1228</sup> See chapter 5, p. 196, and the rule file for domestic architecture, agora area and vegetation (also in Figure 6.13) for a comparison of house plots across sites in mainland Greece.





Figure 6.14 Examples of the procedurally modelled house shapes: Type 2 (top, left); Type 3 (top, right); Type 4, in this instance with closed ground floor (bottom, left), and type 5 (bottom, right). Scan the QR code to view a 3D model of one of the houses on SketchFab.

```
attr porch = 70% : true
            else : false
```

Figure 6.15 Example of how an attribute (in this case 'porch') is used to guide the random, yet controlled generation of 3D geometry. In this case, 70% of the instances will be created with a porch, while the rest will be assigned a closed ground floor.

By comparing the actual appearance of the terrain and the obtained surface created by streets and lots, one notices that the latter does not follow exactly the former, but it approximates it to a degree that depends on several factors, mainly on the amount of nodes that have been introduced in each street and on the size of each lot. It is therefore necessary to create streets that contain several nodes in order to facilitate the alignment between the terrain and the reconstructed environment. It must be noted, however, that in case of hilly sites a certain degree of approximation will be introduced in any case between the original terrain data and the streets/lot layer (see Figure 6.11); it is therefore important to realize this when running analyses and simulations such as those that will be discussed later on in this chapter, as the results will be affected by it.<sup>1229</sup>

### 6.2.3 Rule based modelling of Koroneia's 3D reconstruction

The core of the methodology discussed in this chapter consists of writing a set of CGA rules for the creation of domestic architecture and vegetation, streets and city walls with towers that are used to populate Koroneia's 3D reconstruction. A sample of the rules is available at <https://github.com/cpiccoli/rules>. In the next paragraphs, I will discuss their main characteristics.

<sup>1229</sup> One has to remember in fact that CityEngine's primary application is geodesign, to create several different development scenarios in 3D to communicate with other parties and involve communities in the decision making process, see in this respect the tutorials developed by Geodan and VU University Amsterdam at [http://wiki.bk.tudelft.nl/toi-pedia/CityEngine\\_Open\\_Course](http://wiki.bk.tudelft.nl/toi-pedia/CityEngine_Open_Course) (last accessed March 2017).



Figure 6.16 Top: Examples of the images that I have created for texturing and bump mapping (texture/ bump image for doors; texture for the outer walls). Bottom: Example of rendering (in e-on VUE Infinite 2015 PLE) of the procedurally modelled town with textures applied.

### *Domestic architecture*

The overview of excavated houses that I have presented in chapter 5<sup>1230</sup> has informed the parametric description of domestic architecture in the rule files that I have written for Koroneia's 3D visualization. I have created a series of attributes and constants which precedes the starting point of the rule, by which I controlled the houses' dimensions and the vegetation density (Figures. 6.12 and 6.13). Declaring the attributes at the beginning of the rule allows an easy overview of the available parameters and the quick modification of the numerical values or relationships between attributes, that are then passed down wherever the attribute is mentioned in the rule file. The rule gives moreover the possibility to choose between different configurations of space based on the size of the initial lot, e.g. in regards to the presence and dimension of the courtyard, the extension and appearance of an upper floor (covering the whole surface of the house, only the back, or either side of the courtyard).

I have adopted different modelling strategies to create some variations in the configuration of space and position of the courtyard. In Greek architecture, in fact, the peristyle was introduced towards the transition with the Hellenistic period. I have therefore created an option where the peristyle is present,

<sup>1230</sup> See pp. 194-200.

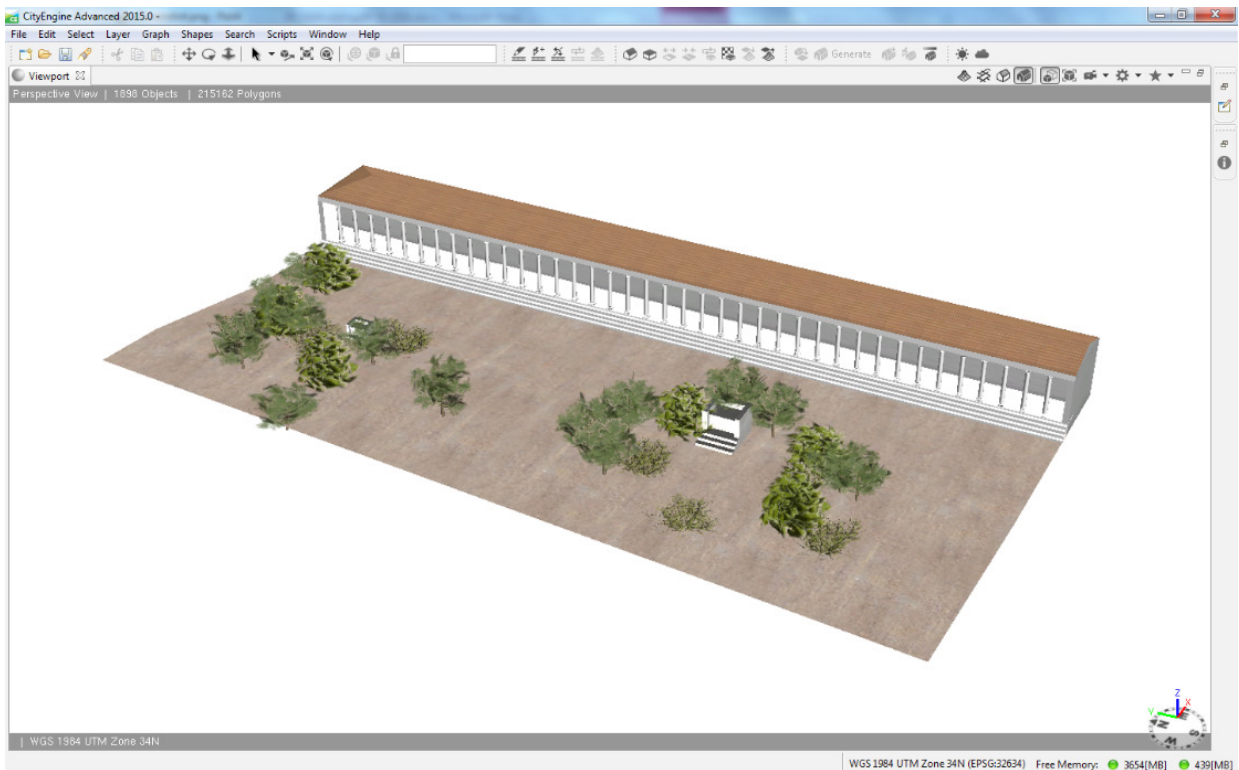


Figure 6.17 The procedurally modelled agora area which is occupied by a long stoa on the eastern side and scattered trees and altars on the open space.

but I had to model also other house types, typically with the courtyard having a porch on some of its sides but not all four of them. The resulting house types have been modelled in 5 different ways (see Figure 6.14):

1. A house without courtyard (by extruding the initial lot in its entirety according to different heights).
2. A two-storey house with in front a courtyard surrounded by one or two wings (by splitting the initial lot in two and modelling each part separately).
3. An L-shaped two-storey house with two sides overlooking the courtyard (using the built-in 'shapeL' operation).
4. A U-shaped two-storey house that surrounds the central courtyard (using the built-in 'shapeU' operation).
5. An O-shaped two-storey house that surrounds the central courtyard (using the built-in 'shapeO' operation).

The different options are generated according to the size of the initial lots: for example, in small lots option 1 is created, while option 4 is assigned to lots that occupy a large area. To further increase the variability among house types, I have included two different possibilities for the ground floor, one which creates a porch (which results in a house similar to the Olynthian standard house, typically House A vii 4) and the other which creates a closed wall and an entrance door. The two options function through the attribute 'porch' which sets the percentage of its presence (Figure 6.15).

The textures that are used to visualize the external appearance of walls, wooden and stone elements, and roofs, were either created in GIMP or re-used from training projects (e.g. the courtyard textures are

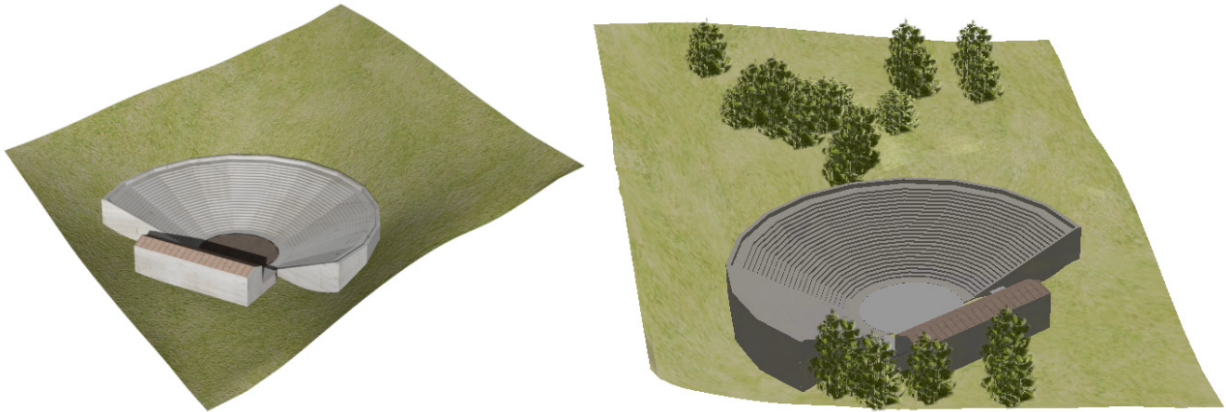


Figure 6.18 The 3D model of the theatre (in OBJ format) that I have created in Blender and imported into the CityEngine using an ad hoc rule. To create some vegetation around the theatre, I have used a scatter operation which randomly places on the surface a set number of 3D models of trees according to a Gaussian distribution. Left: The original 3D model of the theatre as modelled in Blender; right: screenshot from the CityEngine viewport showing the result of the application of the CGA rule file (with vegetation displayed with a high LoD).

taken from the medieval town project, which is part of CityEngine’s training materials)<sup>1231</sup>. For buildings, I have created different images for both texture and bump mapping (which simulates irregularities on the surface) of walls, doors and windows (see Figure 6.16).

### Agora

The CGA code that guides the modelling of the agora is included in the rule for domestic architecture and can be initiated from the so called inspector panel using the tag ‘Agora’. The provisional results of the geophysical prospections seem to show an elongated building bordering the eastern side of the agora, while the rest of the area seems to have been left empty.<sup>1232</sup> Currently I have adopted the interpretation that this long building was indeed a *stoa* and I have therefore modelled the agora with a long *stoa* on its eastern side, leaving the rest as an unbuilt space where some trees and altars are scattered (Figure 6.17). The inclusion of the altars is at the moment based only on Pausanias’ account, which describes the agora in his own times, but confirms the reasonable assumption that there were probably some altars on Koroneia’s agora also in the 4th century BC. Since no archaeobotanical investigations have been carried out on Koroneia’s hill, for the moment I have used 3D models of generic trees, for example plane trees in the agora as they are attested in

```
attr amphiprostyle = true
Temple --> BaseRoof
    Podium t(0,podiumH,0) color( TempleColor )
    TempleTypes

TempleTypes--> case amphiprostyle == true :
    split (x) { 3 : BackTemple
                | ~1 : cella
                | 3 : FrontTemple }
    else : split (x) { ~1 : cella
                    | 2 : FrontTemple }
```

Figure 6.19 Lines added in the rule file for temple architecture to be able to visualize two possible reconstruction hypotheses of the supposed Itonion.

<sup>1231</sup> Available at <http://www.arcgis.com/home/item.html?id=2ac365326981411783b1fa6f0d5e9714> (last accessed March 2017).

<sup>1232</sup> Cf. chapter 4, p. 130.

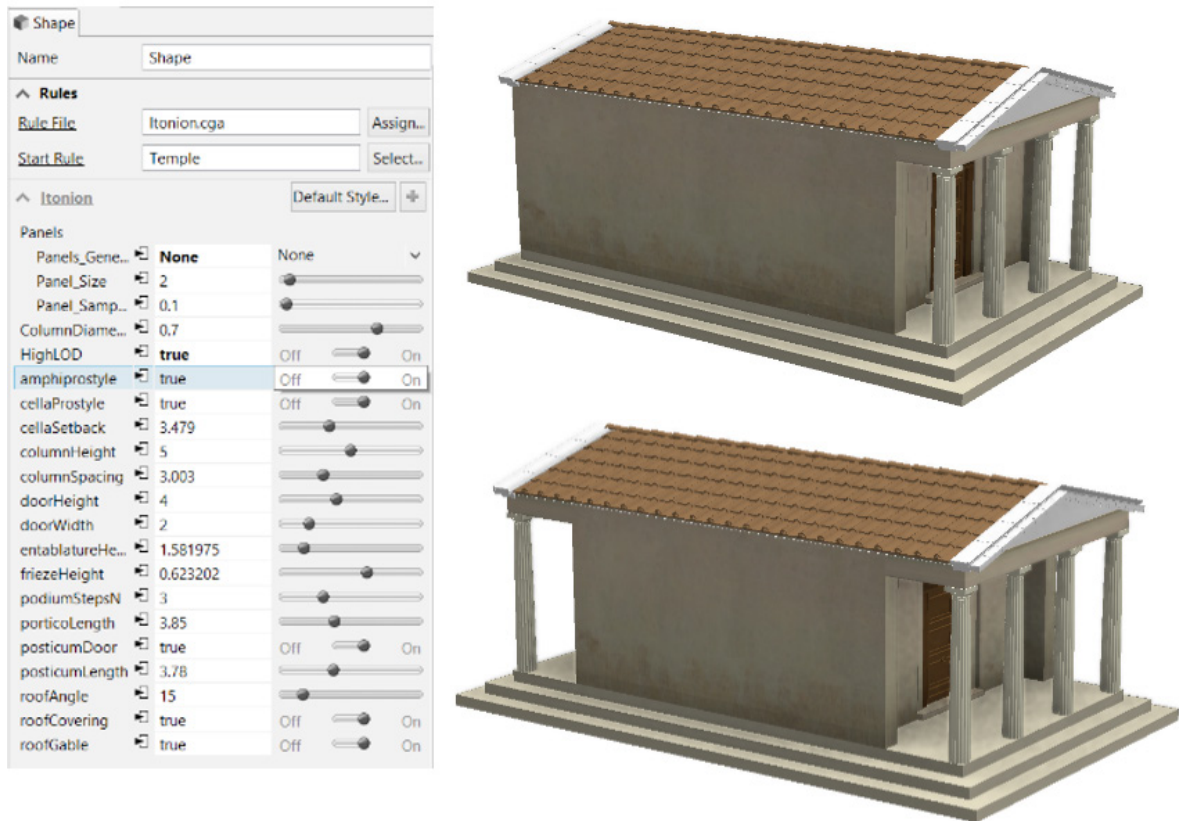


Figure 6.20 Left: Screenshot from the CE inspector with highlighted the Boolean attribute 'amphiprostyle' defining whether columns are only at the front or also at the back of the temple. Right: Visualization of the two reconstruction hypotheses that can be swapped in real time by switching the attribute 'amphiprostyle' on or off.

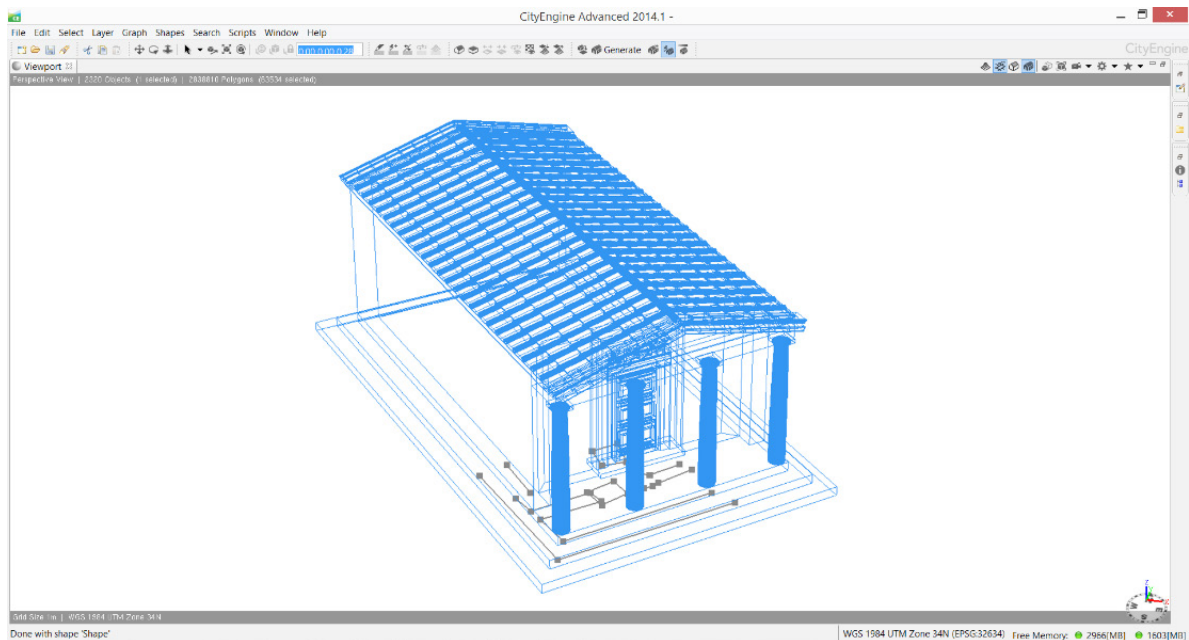


Figure 6.21 The shapefile containing the remains of the temple as recorded in the field (grey lines) is imported into CityEngine and the procedurally generated 3D reconstruction of the temple (here in wireframe mode) is created on top of it, thus making clear the spatial relation between the original data and the reconstruction.

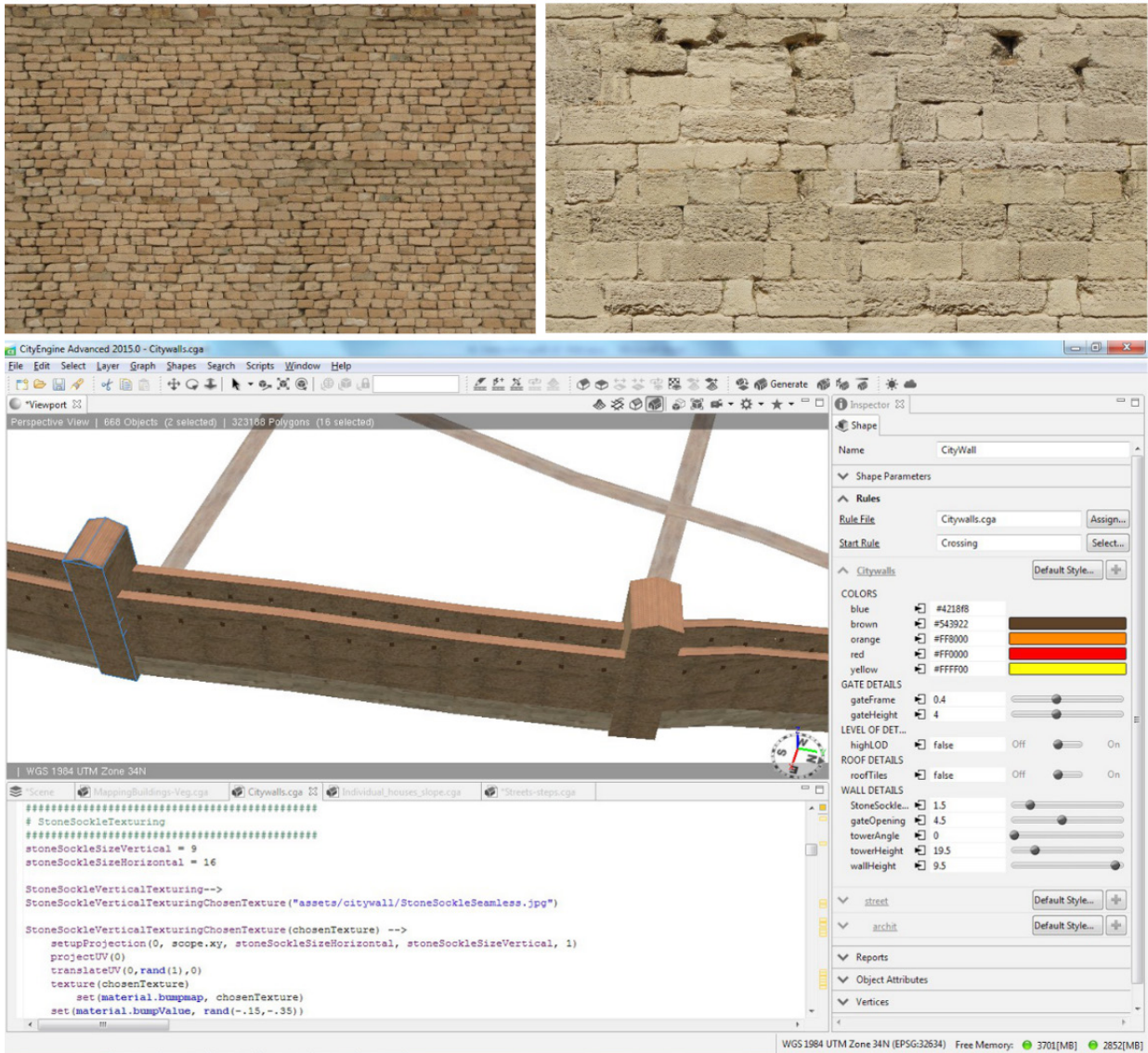


Figure 6.22 Textures of the city wall circuit of the lower town and screenshot from the CityEngine workspace displaying a rule based modelled stretch of the walls. The rule allows the creation of towers at crossing points with streets by inserting ‘Crossing’ in the inspector’s field ‘Start Rule’.

ancient sources,<sup>1233</sup> which have been made freely available as part of the Esri/LumenRT vegetation library.<sup>1234</sup> In the future, if more data will be available, these generic trees can be easily substituted by modifying the file path of the current 3D models in the rule file, with the file name of the updated 3D trees displaying a more specific type of vegetation. For the modelling of the altars, I have relied on a manual modelling approach using SketchUp as the time spent for modelling such elements with the CGA shape grammar would not have been justified given the fact that they were individual objects.

<sup>1233</sup> See chapter 5, p. 221.

<sup>1234</sup> The library is available at <https://www.arcgis.com/home/item.html?id=0fd3bbe496c14844968011332f9f39b7> (last accessed Oct. 2016).

### Theatre

The theatre was manually modelled using the open source software Blender. As there is no archaeological evidence hinting at the overall appearance of the theatre, I chose to model a simple construction with a stone *skene* whose dimensions fit the depression it left on the hill.<sup>1235</sup> The insertion of the theatre posed the additional challenge of finding the best solution to place the 3D model of the theatre despite the uneven morphology of the terrain. This would have caused in fact either the initial lot to intrude in the volume of the theatre or the theatre floating on the lot's surface. To overcome this problem, I modelled both the theatre and its surrounding terrain directly in Blender and substituted the resulting 3D model to the lot corresponding to the theatre's position. Since this strategy required a very precise fitting between the edges of the 3D model and the surrounding streets, I have exported the four streets that surrounded the theatre lot from CityEngine in DAE Collada format and imported them in Blender in order to model the terrain according to its precise shape. I then exported the thus obtained 3D model as an OBJ file and inserted it in the CGA rule file for the theatre. I have moreover included some vegetation around the theatre by using some of the 3D models of the free Esri/LumenRT vegetation library (Figure 6.18).

### Temple architecture

The rule file for temple architecture is a modification of the rule file 'parthenon.cga', which was written by Pascal Müller and it is available as learning material for the CityEngine software.<sup>1236</sup> I have maintained the overall structure of the original rule and reused the assets that were included in it (e.g. 3D models of columns, capitals, roof tiles), but I modified some parts and added some lines of code to display alternative reconstruction hypotheses for the temple architecture. Specifically, I have defined the attribute 'amphiprostyle', which can be switched on or off to display either a temple with the portico only on the front (*prostyle*), or an *amphiprostyle* temple, with a portico both on the front and on the back (Figures 6.19 and 6.20). I have applied this rule to suggest two alternative reconstruction hypotheses of the temple located in the plain just north-east of the hill and partially excavated in the 1970s by Spyropoulos.<sup>1237</sup> The front portion of the temple which was unearthed during the excavations does not reveal the characteristics of the back, hence the idea to include the attribute 'amphiprostyle' in the rule file. I have imported in CityEngine the shapefile containing the outline of the excavated remains as recorded in the field in 2012 in order to create the 3D reconstruction of the temple on top of the GIS data. The aim of this approach was to provide an intellectually transparent visualization with a clear distinction between original data and interpretation (Figure 6.21).

### City walls

I have created the city walls using the same tool that creates the street network, which allows the editing in real time of their position by dragging the nodes that compose the graph. In this way the circuit can easily be adapted to the terrain and can be modified if new data would alter our hypothesis regarding its path. To create the rules for the city walls I have taken as the starting point the rule for the city walls in the CityEngine show-case project 'Medieval town' which I have modified to suit our purposes. The rule is constructed in such a way that towers are created at crossing points of the city wall circuit, which is a convenient strategy to introduce new towers at the target location. As discussed in chapter 4, so far, in fact, the architectural survey at Koroneia has only yielded evidence for two

<sup>1235</sup> See chapter 4, 132 and chapter 5, § 5.3.3 for comparisons.

<sup>1236</sup> The rule is part of the Tutorial 09 Advanced Shape Grammar: Parthenon available at <https://www.arcgis.com/home/item.html?id=47c0cc67406641298c10b720415c1fa5> (last accessed Oct. 2016).

<sup>1237</sup> See chapter 4, 103-5.

```

// Functions to determine orientation, slope
y_angle = convert(y, pivot, world, orient,0,0,0)

setpivot_index_uncorrected = floor((y_angle+45)/90)
setpivot_index0 = setpivot_index_uncorrected+0 - floor((setpivot_index_uncorrected+0)/4)*4
setpivot_index1 = setpivot_index_uncorrected+1 - floor((setpivot_index_uncorrected+1)/4)*4
setpivot_index2 = setpivot_index_uncorrected+2 - floor((setpivot_index_uncorrected+2)/4)*4
setpivot_index3 = setpivot_index_uncorrected+3 - floor((setpivot_index_uncorrected+3)/4)*4

slopeDegrees (sx, sy, sz) = geometry.angle(maxSlope)

// START RULE
@StartRule

LotInner --> Lot

Lot --> alignScopeToAxes (y)
    Lot1

/*
From "The Neighborhood Model: Building Block for the Development Areas", p. 106:
On slopes 0 - 10% (0 - 5.74°): most building can occur easily without major regrading
On slopes 10 - 15% (5.74° - ca. 9°): moderate grading may be needed to support development
On slopes 15 - 25% (ca. 9° - ca. 14°): more care with grading is needed so that extremely
steep slopes are not created by regrading
On slopes 25% and greater: extreme care should be taken when building on slopes of 25% so
slopes that are even steeper than 25% are not created. Note however that archaeological
evidence shows that hilly terrain were exploited for construction up to very steep slopes
such as at Ephesus, where the preferred slope was up to 30° of gradient (Groh 2012, 68).
[at Petres of Florinas the hill's gradient was 30-45% (ca. 17° - 24°; see P. Adam-Veleni
2000, Petres of Florinas, 35 - the maximum gradient for construction on slope however is
not given)].
*/
Lot1 -->
    case slopeDegrees (scope.sx, scope.sy, scope.sz) > 13 :
        report ("Area.Greenspace", geometry.area)
        LotVegetation

    else:
        alignScopeToAxes (y)
        innerRectangle (edge)
        {shape:
            t (0, -2, 0)
            extrude (world.up, 2)
            comp (f)
            {top:
                alignScopeToGeometry (yUp, 0)
                alignScopeToAxes (y)
                setPivot (xyz, setpivot_index1)
                // in this case I chose the N-E corner of the shape's scope as target pivot
                BuildingArea
            |side:
                alignScopeToGeometry (yUp, 0)
                TextureAgora
            }
        |remainder:
            TextureAgora
        }
}

```

Figure 6.23 Beginning of the annotated rule file for the modelling of Koroneia's houses, showing the lines that allow setting the pivot of the initial shape to a defined location (see section 6.2.3 for a more detailed explanation of this work-around). In this case I chose the north-east corner of the shape's scope as target pivot to be able to orient the courtyards towards south and east. The rule includes moreover the function to calculate the slope, which informs the generation of the 3D environment and has been used to calculate the number of houses in relation to different slope input values (see below, § 6.3.2).



```

/*
Cf. streets's width in 4th century BC sites:
BOEOTIA:
Plataia= street connecting West and East tower (main street): 6.5m; acropolis area: 4.20-
6.70m; South of acropolis: 4.50m-5.50m; East of acropolis: 4.90m-5.80 (Konency 2013, 123-
125)
Tanagra= running NE-SW: ca. 3m (10ft); running SE-NW: ca. 5m (15ft) (Slapsak 2012, 56)
THESSALY:
Halos='Main avenue': 9.11m; Avenues A,B,C,D = 6.5-7.5m; others: 5.5m Reinders 1988, 208)
Goritsa= width varies along the street: major streets: average 4.95m; others: 3.25m (see
Bakhuizen 1992, 171-245)
CHALKIDIKI:
Olynthos= Hoepfner and Schwandner 1994, 77: 5.05m; From Cahill 2002, ill. p. 28: ca. 5m and
ca. 7.35m
PELOPONNESE:
Argolis/Halieis= Avenue A (underwater); Avenue B: 3.55-4.20m; Avenue C: 4.50-5.05m;
crossing roads: 2.60-2.80m (Boyd-Rudolph 1978, 339-345)
Messenia/Messene= various width: from 2m to 12.6m (Müth 2007, 245-256 and table p. 274)
Arkadia/Kyparissia= six parallel streets: 4.60m; plateiai not found (traces seem to
indicate a width of 8m) (Karapanagiotou 2005, 333)
EPIRUS:
Kassope= main street: 6m; 19 narrower streets: 4.50m (Schwandner in Isanger 2001, 109)
ASIA MINOR:
Ionia/Priene= 'Quellenthorstrasse': 5.60-6.18m; 'Athenastrasse': 3.64-4.35m;
'Westthorstrasse': 7-7.35; other streets: 3.10-5.60m (Wiegand and Schrader 1904, 48)
Ionia/Ephesos (3rd BC)= processional way: 8m; plataiai (E-W): 5m; stenopoi (N-W): 3m (Groh
2012, 69)
*/

@StartRule
Street --> alignScopeToAxes ()
    Lot2
Lot2-->   case convert(y, scope, world, pos, 0,1,0) - convert(y, scope, world, pos, 0,0,0)
    < 0 : reverseNormals
    print (slopeValueDegreesStreet)
    LotNormalUp
    else: print (slopeValueDegreesStreet)
    LotNormalUp

slopeValueDegreesStreet = atan (scope.sy / ( scope.sx * float(scope.sx > scope.sz) +
scope.sz * float(scope.sx <= scope.sz) ))

LotNormalUp -->
    case slopeValueDegreesStreet > 11 && slopeValueDegreesStreet < 14: Steps(0.30)
    case slopeValueDegreesStreet > 14 && slopeValueDegreesStreet < 18: Steps(0.25)
    case slopeValueDegreesStreet > 18 && slopeValueDegreesStreet < 27: Steps(0.20)
    case slopeValueDegreesStreet > 27 : NIL
    else: setPivot (xyz,1)Street1
    print (slopeValueDegreesStreet)
/*
From "Footpaths and Tracks. A Field Manual for their Construction and Improvement",
2002, p. 3-26:
Desiderable maximum gradient:
For animal drawn carts: 8% (4.6°);
For pedestrians and packed animals: 12% (6.8°);
as slope approaches 25% (14°) walking becomes difficult and steps are needed:
Shallow steps required: 25%
Moderate steps required: 33% (18.3°)
Steep steps required: 50% (26.6°)
*/

```

Figure 6.24 Beginning lines of the CGA rule file that is applied to streets. The file starts with commented out references to the dimensions of streets in other 4th century BC sites that have been excavated or surveyed in Boeotia and elsewhere. The proper rule starts at @StartRule and includes the formula to calculate the slope degrees and the conditional rule (case ... else) that guides the creation of a street or different types of steps within the set range of slope degrees. For reference, I have included as a comment the indications from modern construction guidelines.

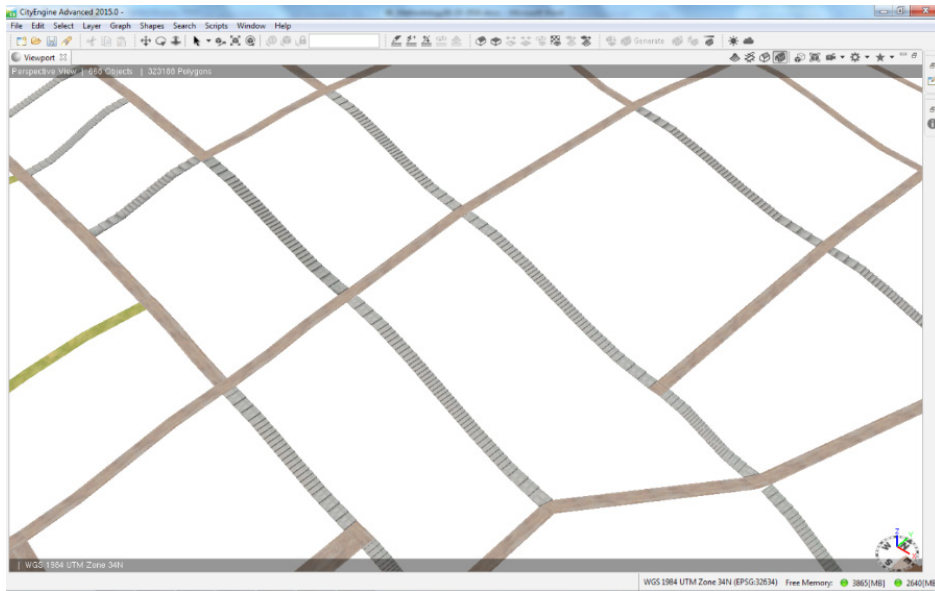


Figure 6.25 Screenshot from the CityEngine workspace where only the street network is selected, to show the dynamic rendering of streets and steps according to the slope degree by applying the rule file written for this project.

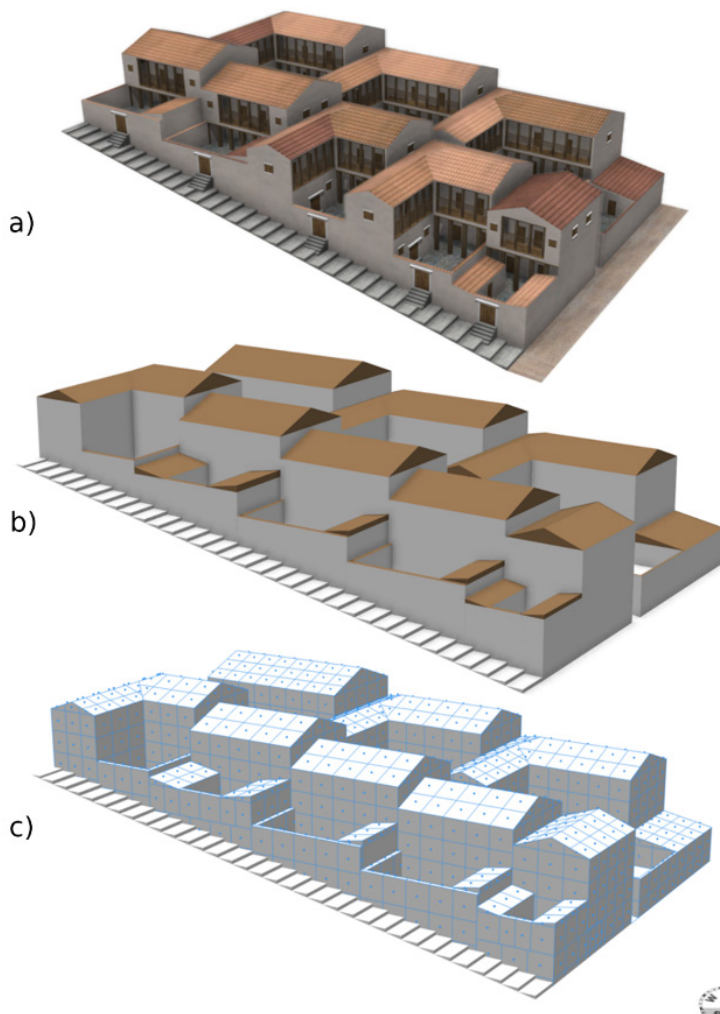


Figure 6.26 A city block generated by the procedural rule for domestic architecture; in a) one of the possible different configurations of space that are encoded in the rule file are represented (HighLOD = true); b) displays the low Level of Detail scene, which generates geometries as coloured volumes (HighLOD = false); c) shows the panels and points that can be exported and used to perform visibility analysis in ArcGIS, as shown in Figure 6.29.

towers on the western slope but it is likely that geophysical prospections will provide further indication for the position of other towers.

The rule generates a polygonal masonry for the acropolis walls, as attested by the stretches that were found during the survey at Koroneia, and a regular, rectangular stone foundation with its higher elevation in mudbrick for the lower circuit wall, as this is the most common construction technique for Classical walls. The various building techniques are visualized by using different textures which I have created in the open source image editor GIMP. These textures originate from images available on the web which I have modified to make them seamless, which avoids the creation of a noticeable edge when the texture is repeated several times to cover the entire surface (see Figure 6.22).

### *Slope dependency*

In the case of Koroneia, the hill's morphology must have conditioned the building choices, as is emerging from the currently available survey results and comparisons with other Boeotian sites.<sup>1238</sup> For this reason, the slope variable needed to be included in the 3D visualization, to be able to create different scenarios by interactively changing the value set as maximum slope degree allowed for building. To this end, I have inserted three possibilities in the rule file for domestic architecture: the calculation of the slope degrees of the initial shape at the beginning of the rules, the definition of a threshold for the maximum slope allowed for construction, and the report on entity count that allows the automatic calculation of the generated number of houses (Figure 6.23). In this case study, one of the biggest challenges was the modelling of buildings with an open courtyard on hilly terrain, which made it necessary to compensate for the sloping ground and avoid that the terrain would protrude through the courtyard (see section 'rule-based modelling: challenges and workarounds' for additional information).

Similar to the rule for domestic architecture, also the rule that I wrote to texture the street network includes the calculation of the slope. The rule works in such a way that steps are created instead of a street when the slope becomes too steep to ascend (see Figures 6.24 and 6.25). Differently than for

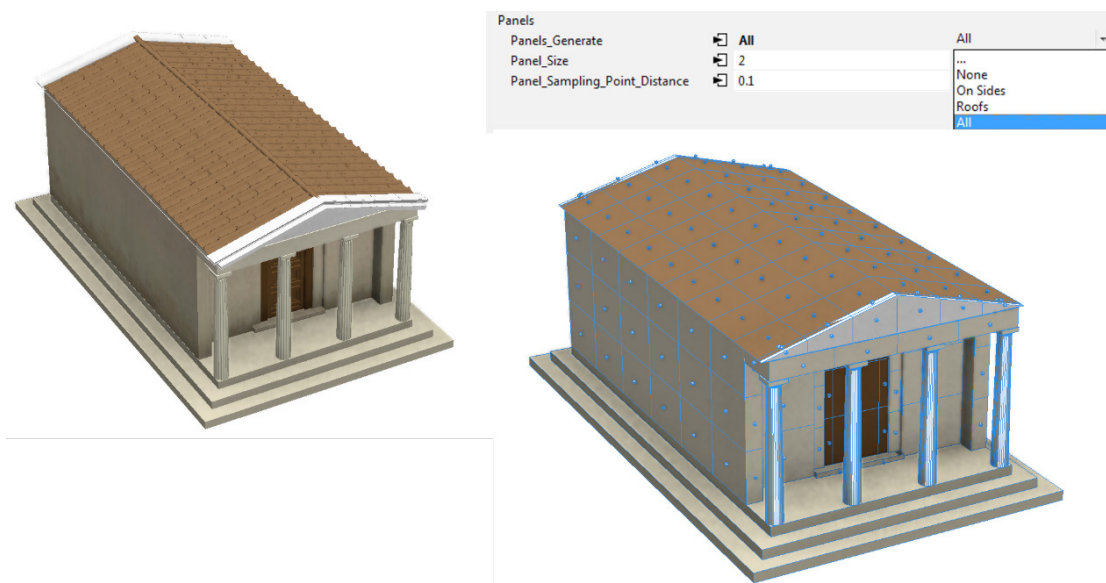


Figure 6.27 The procedure of sampling points and panels on the building's walls and roofs is triggered by switching between the options of the attribute Panels-Generate in the inspector.

<sup>1238</sup> E.g. Hyettos and Haliartos, see p. 132.

the domestic architecture, the slope of the street segment is calculated with the arctangent function using the longest side of the lot as adjacent side, since this corresponds to the direction in which the street will be traversed and, hence, to which the slope applies. The largest value of two variables A and B was determined by the function:  $A * \text{float}(A > B) + B * \text{float}(A \leq B)$ . Regarding the values that I have used as threshold, comparisons with the Roman world show that roads built by the Romans never exceed 15%, which corresponds to  $8.5^\circ$ .<sup>1239</sup> This value generally agrees with modern construction guidelines, where the desirable maximum gradient for animal drawn carts is given at 8% ( $4.6^\circ$ ), for pedestrians and packed animals is 12% ( $6.8^\circ$ ), while as slope approaches 25% ( $14.0^\circ$ ) walking becomes difficult and steps of different depths are needed (25%: shallow steps required; 33% ( $18.3^\circ$ ): moderate steps required; 50% ( $26.6^\circ$ ): steep steps required).<sup>1240</sup> The rule includes moreover some comments on the dimensions of streets that have been excavated or surveyed in 4th century Greek sites and the indications from modern construction guidelines as a comparisons for the values that I have inserted as thresholds (see Figure 6.24). The insertion of these indications contribute to the development of an intellectually transparent 3D visualization.

### Level of Detail

Two attributes ('LOD' and 'HighLOD') allow the control of the Level of Detail of vegetation and buildings respectively. The choice of having two different attributes to control the LoD, which can be operated independently, is justified by the easiness to create a different mix of details in the scene that can be changed rapidly when needed (e.g. high LoD for buildings and low LoD for vegetation). By setting the HighLOD attribute as 'true', detailed geometries are created, but roofs are kept as textures to save on the polygon count. When the HighLOD attribute is 'false', instead, the geometries are created with low detail as schematic coloured volumes (Figure 6.26 a, b). I have inserted this second option to keep the

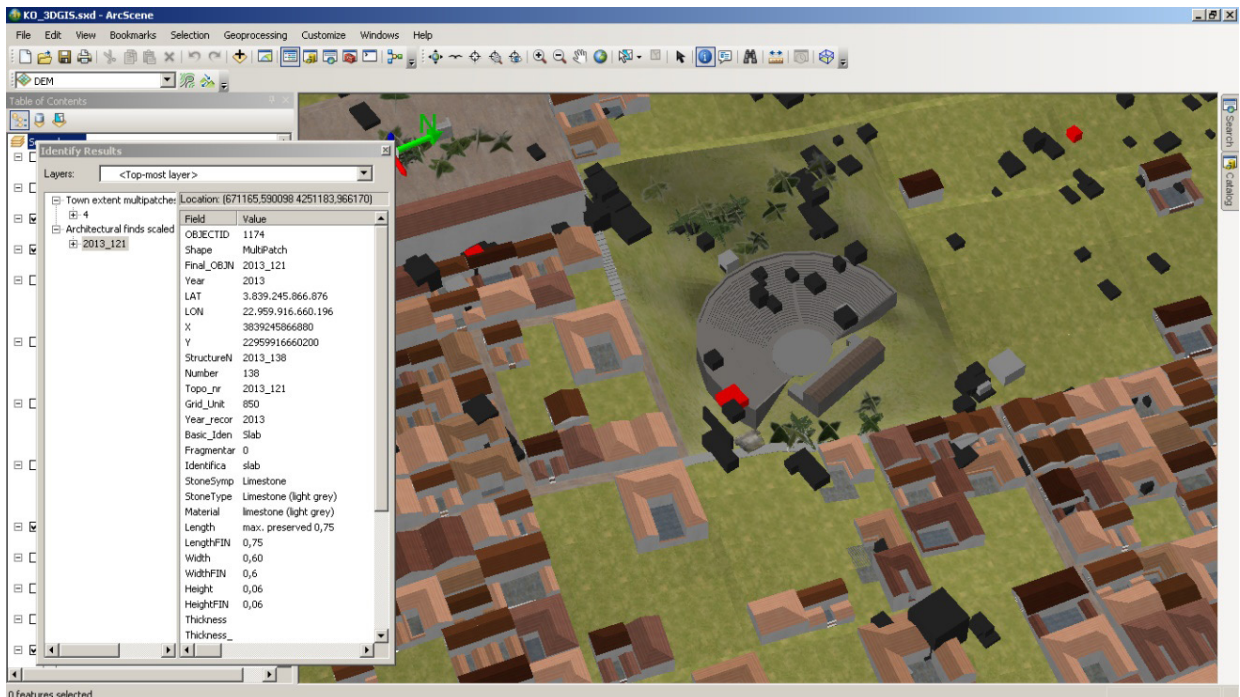


Figure 6.28 Screenshot from ArcScene workspace showing the overlay between the architectural data displayed using the procedural rule described above and the procedurally modelled reconstruction hypothesis (in low LoD) imported as gdb.

<sup>1239</sup> Verhagen *et al.* 2014, 80 citing Hucker 2009 and Quilici 1995.

<sup>1240</sup> I.T. TRANSPORT Ltd. 2002, 3-26.

polygon count at a minimum for exporting the whole scene to another software when only the volumetric information is needed (e.g. for analytical purposes in ArcGIS or for setting up the scene in a CG software). Two different levels of detail have been created also for vegetation, from the LoD of architecture. A low LoD creates trees as a cross-billboard, which consist of two crossing images displaying the tree with a transparent background, and a more detailed version that can be used for a more realistic visualization.

### *Sampling points on buildings' surfaces*

In the rules for domestic and public architecture I have included the possibility to sample points and create panels on the buildings' surfaces. By inserting at the right locations the lines of codes within the rule file, and initiating the corresponding attribute in the inspector (in this case *panels: on*), it is in fact possible to divide the building's facades in equal squares and to create a point at each square's centroid (see Figure 6.26, c and 6.27).<sup>1241</sup> This workflow has been recently introduced for the spatial and visual impact analysis of modern cities,<sup>1242</sup> and it is based on the possibility to export each of the so called leaf-shapes that are generated by a rule as individual GIS features, so that they can be used to perform visibility analysis on the built environment in ArcGIS. Points are used to run a Line of Sight analysis and panels can be used to map the frequency count in order to visualize the visibility of the portion of the building defined by each panel (see below, section 6.3.1).

## **6.3. Results**

The methodology described above has allowed us to create a 3D GIS of the architectural survey data which exposes in an intuitive way some characteristics of the dataset (see section 6.2.1), and to create alternative reconstruction hypotheses of Koroneia's town layout which reflect the current state of our data. The combination of these two layers of information in a 3D GIS allows a visual and spatial overlap between the survey data and the reconstruction hypothesis, thus allowing the possibility to store and compare data and interpretation within the same platform (Figure 6.28).

As will be discussed in this section, the procedurally modelled 3D reconstructions can then be used for a variety of analytical purposes, such as to compute the degree of integration of the street network, to run a visibility analysis in ArcScene, and to estimate population figures according to different input parameters.

### **6.3.1 Visibility analysis in a 3D GIS**

Traditionally, visibility analysis on 3D models in GIS has been problematic. In 1997, Esri developed the multipatch geometry type to define the exterior shell of 3D objects in GIS. Besides the volumetric properties of the object, this format can also store texture information.<sup>1243</sup> Although the implementation of this geometry type pushed considerably forward the capabilities of GIS to handle 3D models, the problem remained that each planar surface that composed the 3D geometry could not be subdivided in smaller subsets. This limited considerably the possibilities for visibility analysis on the built environment, as a building could be either seen in its entirety or not seen at all.<sup>1244</sup> To perform a visibility analysis of the built environment in a 3D GIS, four main elements need in fact to be specified: a terrain layer, an observer, a target, and an obstacle. In the first phase, sightlines are created between the observer point(s) and the target point(s) using the 'construct sight lines' tool in the 3D Analyst toolbox. Next, the line of sight is calculated including the terrain layer and the obstacle layer (e.g. other buildings that

<sup>1241</sup> The sample rule files can be downloaded from <http://www.arcgis.com/home/item.html?id=b7142081948b4cd9ac1c852d8790c9c9>

<sup>1242</sup> Van Maren 2014.

<sup>1243</sup> See 'The Multipatch Geometry Type. An Esri White Paper – December 2008', Esri 2008 available at <http://support.esri.com/en/knowledgebase/whitepapers/download/fileid/5111>.

<sup>1244</sup> For a thorough discussion on visibility analysis in the fields of urban and landscape research see Paliou 2013.

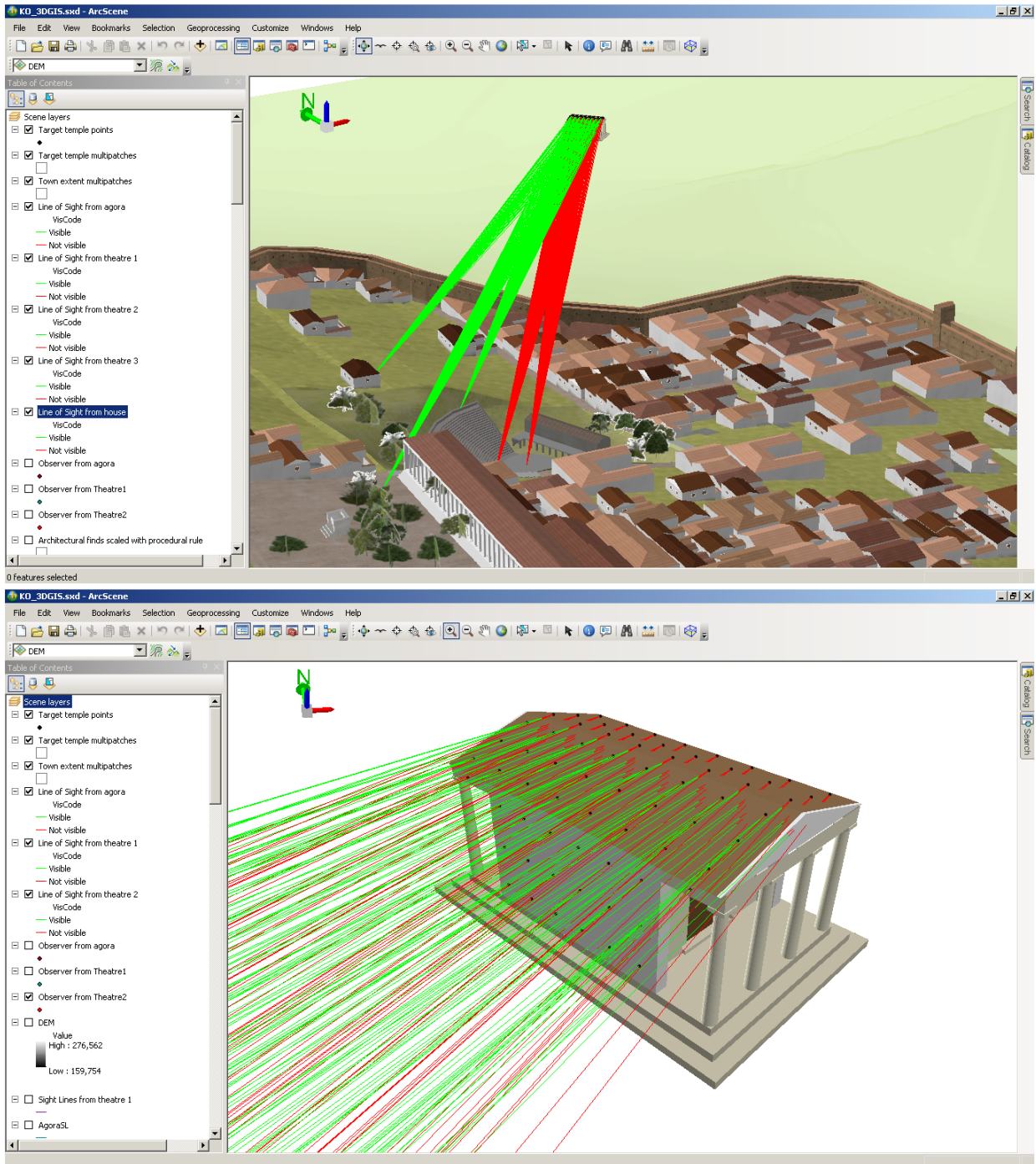


Figure 6.29 The application of the possibility to sample points on the buildings' facades. Top: the results of the Line of Sight analysis that was run in ArcScene on a portion of the procedurally modelled environment in CityEngine to map which parts of the temple are visible from the observer points that were located on the agora, on the theatre and in other parts of the lower town. Bottom: A close-up of the temple which shows the sight lines coded according to the target points' visibility from the observers.

might obstruct the line of sight between observer and target) using the 'Line of Sight' tool in the 3D Analyst toolbox to distinguish between the sight lines which hit the target and those that were instead blocked by the elements in the obstacle layer.

The method that is adopted here allows the automatic sampling of points over the building's planar surfaces and their subdivision into panels and has been developed by ESRI as part of the workflow for the analysis of the 3D modern city.<sup>1245</sup> As discussed above, I have in fact included in the rule file for Koroneia's project the possibility to sample points and panels on the buildings' facades (see Figure 6.26 and 6.27). This integration makes it possible to switch between three different visualization modalities (photorealistic, volumetric and panels/points) within the same rule file. The geometry thus created is exported from CityEngine as a geodatabase by choosing as the feature mapping mode the 'one feature per leaf shape' option in the export dialog. In this way, the points are saved as a point feature class, while the panels are stored as a multipatch feature class and can be therefore used for analysis in ArcGIS. (A portion of) the thus obtained points can be used as observer or target in the intervisibility analysis, while the panels can be used to map the frequency count in order to visualize the visibility of the portion of the building defined by each panel. For Koroneia, we use this workflow to evaluate the visual prominence of the temple partially excavated by Spyropoulos north-east of the hill. As discussed in chapter 4, finds on this area show that the temple was part of a larger sanctuary which was in use from the Archaic to the Late Roman period. It must therefore have been an important cult centre, which some scholars have identified with the regional sanctuary of Athena Itonia (but see chapter 4 for this debated interpretation). Based on the currently available data and with the citywall reconstructed as 8 meters high (an average within the range indicated by Winter for Classical city walls),<sup>1246</sup> the temple would have been visible from some parts of the agora and the upper town, but not visible from most seats in the theatre (Figure 6.29). This method can be taken further in case additional geophysical prospections performed in the area of the sanctuary and of the theatre will clarify the extension of the former and identify the location of the *skene* of the latter, thus giving a more precise indication of its orientation. For example, it would be interesting to create different hypotheses about the height of the city walls and see what impact they would have had on the visibility of the sanctuary in the plain from the town's centre.

The creation of a grid of points overlaid on a wall or facade allows parts of the buildings to be considered as individual entities, thus opening up the possibility to use the 3D visualization as an analytical tool to investigate the spatial relationships within and between buildings in a quantitative way.<sup>1247</sup> By using this method, a building can be analysed in its context, to evaluate its presence in relation to its surroundings, while its visibility can be determined from different locations. The ESRI rule that has been included in our rule set allows an easier creation of such a grid; moreover, the use of this method coupled with the parametric modelling of different hypotheses can offer a way to evaluate and analyse alternative scenarios (for example, which height a building should have reached in order to be seen from a given location). As already stressed, in fact, this modelling strategy allows the quick creation of different reconstruction hypotheses simply by changing the defined numerical parameters (e.g. the height of a building), which can be then exported into ArcGIS to test the most plausible reconstruction hypothesis. It must be noted, however, that the export of the 3D scene to the geodatabase may result in very large files. For instance, Koroneia's scene was about 13 GB when exported with a low LoD. This not only results in long export and import sessions, but also in computationally heavy analyses. It is therefore advisable to subdivide the 3D scene in portions and export and import each of the sub-sections individually.

### 6.3.2 Estimating number of houses and population figures

As discussed above, we were moreover interested in developing an analytical visualization that exploited the hill slope as a threshold for the generation of 3D geometry. In this way, we were able to produce different reconstruction hypotheses where we could see how much the number of houses that could fit

<sup>1245</sup> Available here: <http://www.arcgis.com/home/item.html?id=84a432d1b26d4fbfac3b0feb3359fa12> (last accessed Dec. 2016).

<sup>1246</sup> See chapter 5, 217.

<sup>1247</sup> See Landeschi *et al.* 2016 for an application of visual analysis in the 3D GIS of the house of Caecilius Iucundus in Pompeii.

within the city wall circuit would vary, when using the slope threshold as a variable. The final aim of this experiment was to formulate some hypotheses on Koroneia's population size.

The calculation of the inhabitants of ancient Greek *poleis* has triggered the interest of many scholars, starting with Julius Beloch's *Die Bevölkerung der griechisch-römischen Welt* published in 1886, as the amount of population within a territory has far reaching implications for understanding economic strategies and can explain conflicts between neighbouring city states under the pressure of finding new resources. In the following paragraphs, I will discuss some of the traditional methods that have been employed to calculate population figures on an individual *polis* and apply them to Koroneia, to allow a comparison with the numbers that resulted from our 3D visualization.<sup>1248</sup>

Surviving evidence on army figures has been extensively used to infer population size on the assumption that the number of troops sent to fight was proportional to the number of citizens.<sup>1249</sup> For the beginning of the 4th century, we have in fact information about the military organization of the Boeotian League from a fragment of papyrus found at Oxyrhynchus. The surviving lines inform us that in 395 BC, Boeotia was divided in 11 districts, each one contributing a federal magistrate (Boeotarch), and military forces corresponding to 1,000 hoplites (heavy infantry) and 100 cavalry (*Hell. Oxy.* 16.4 [Bartoletti]). Among the Boeotian cities, Thebes for example supplied four Boeotarchs, Orchomenos and Thespias two, Tanagra one, while Haliartos, Lebadeia and Koroneia supplied one in turn.

Using this method to infer the total population presents however a series of problems. If it is certainly true that larger towns must have contributed more than smaller ones to the army, it is not clear what the correct ratio between the two would have been. We must also consider that the amount of light-armed troops and seamen are unknown as they are not included in the papyrus' figures, and that not all men must have served in the army, as some must have stayed to carry the *polis*' administrative tasks.<sup>1250</sup> One has to note, moreover, that it is not certain that the figures that are reported in ancient sources correspond to the real ones, as literary evidence points towards the use of inflated figures.<sup>1251</sup> In his calculation to estimate the population of ancient Boeotia, Bintliff uses these numbers (corrected to include light-armed troops and seamen), and multiplies them by 5 obtaining a total of 165,500 people, including slaves.<sup>1252</sup> If we would apply this method to Koroneia, knowing that this *polis* contributed 1/33 of the total armed forces of the region according to the papyrus' fragment, we would reach a total population of some 5,015 people. In this case, this number would represent both the intramural population and the people in rural sites within Koroneia's *chora*.

An alternative method uses the physical remains of a walled city multiplying the hectares per average number of persons per domestic hectare. For his calculations on the population of some individual Boeotian *poleis* and of ancient Boeotia in general, Bintliff relies on the assumption that about 56% of towns was occupied by domestic units, while the rest was destined to public, commercial and religious functions; he then takes the estimation of 225 people per domestic hectare, which Jameson advanced

<sup>1248</sup> For the various approaches that have been used for calculating population figures in Greece, see the overview in Hansen 2006, 1-34.

<sup>1249</sup> Beloch 1886; see also Bintliff's estimation based on the army figures of the Boeotian Confederacy contained in the Oxyrhynchus papyrus (Bintliff 1985, 141-2, updated and revised in Bintliff 1997).

<sup>1250</sup> For the calculation of the army forces that are not included in the papyrus' figures, see Bintliff 1985, 141-2; for the problems related to the use of army figures, see Hansen 2006, 5-6.

<sup>1251</sup> In this regard, P. Hunt cites the words that according to Thucydides Alcibiades pronounced on the eve of the Sicilian expedition (Thuc. 6.17.5): 'As for their hoplites, they have not got so many as they boast of: it is the same with them as with the rest of the Hellenes; the number never came up to the estimate made by each state of its own power; in fact the falsification was a very big one.' (Hunt 2007). On the other hand, however, it is equally possible that this speech could have been aimed at playing down the enemy forces.

<sup>1252</sup> Bintliff 1985, 141-2.



for Argolis,<sup>1253</sup> and multiplies by the amount of domestic hectares that he had calculated for Boeotia to reach an estimation of the total urban population of the region of 131,985 people and applies this method also to individual towns.<sup>1254</sup> Based on these figures, Koroneia's extent of about 34 (yellow line in Figure 4.32) and 37 (orange line in Figure 4.32) hectares would lead to 19 - 20.7 domestic hectares which multiplied by 225 gives a population of between 4,275 and 4,660 within the walled city. However, it must be noted that the 56% of Greek towns being domestic is derived from the situation at Athens, as described by Travlos in his 1960s publication,<sup>1255</sup> but the proportion between the area within the city walls and the area for housing greatly varies among Greek *poleis*,<sup>1256</sup> the terrain morphology playing a major role in defining the zones suitable for building.

In his 'shotgun method', Hansen adopts different percentages based on the size of the *polis*: for very small *poleis* (of up to 10 ha), he assumes that the habitation area took up around two-thirds of the intramural space, for medium *poleis* (10-150 ha) this area would be about half of the entire intramural space and for large *poleis* only one third.<sup>1257</sup> He then takes an average of 30-33 houses per hectare,<sup>1258</sup> and a minimum average household size of 5.5 people (including slaves),<sup>1259</sup> taking into account the many problematic aspects when estimating the occupants of a house, as e.g. many variations may occur over its lifecycle,

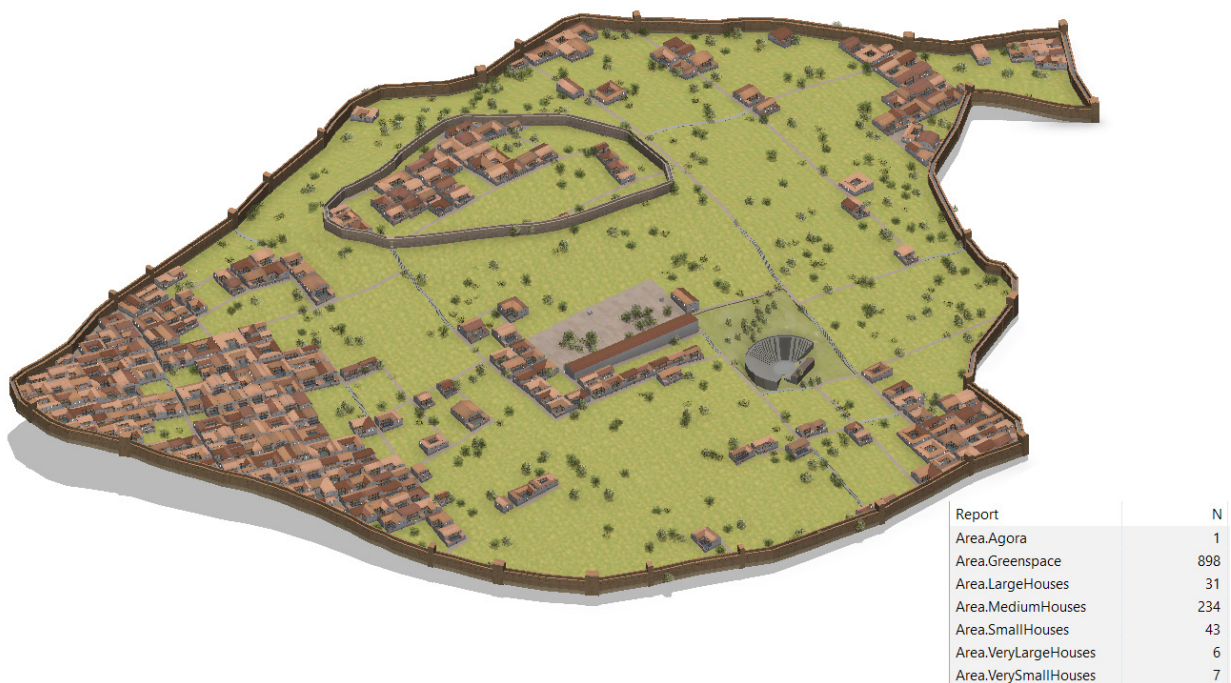


Figure 6.30 Hypothesis 1: The threshold for the terrain suitable for buildings is up to 9°. This scenario returns a total amount of 321 houses (automatically calculated by including the report operation in the rule).

<sup>1253</sup> For other numbers adopted to estimate population figures see also Cahill 2002, note 43.

<sup>1254</sup> Bintliff 1997.

<sup>1255</sup> Travlos 1960, 71; Kolb 1984.

<sup>1256</sup> Hansen 2006, 37-47.

<sup>1257</sup> Hansen 2006, 46-7.

<sup>1258</sup> Among the sites that he lists in table 2.3, the number of houses per ha varies between 26 and 50, but most had 30 to 33 houses per ha (Hansen 2006, 51).

<sup>1259</sup> Hansen 2008, 280; Bintliff reaches a similar figure of 5 people per household, assuming an equal number of men and women, two children per family and a slave (Bintliff 1985, 142).

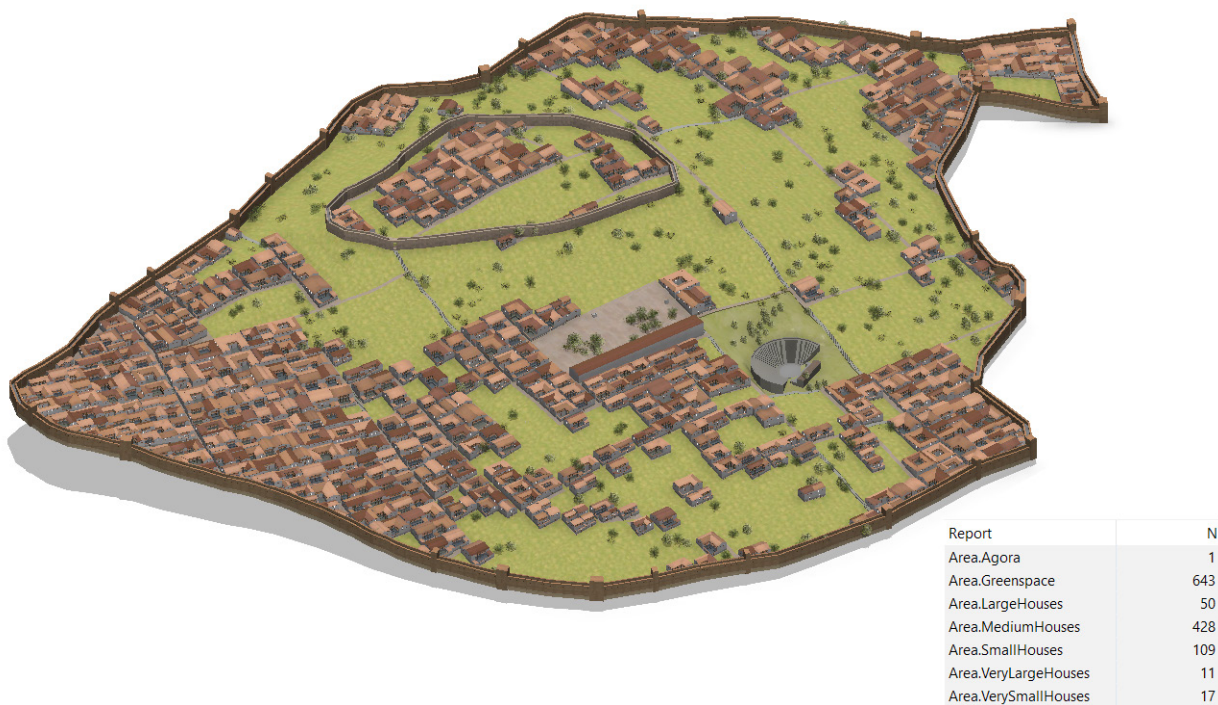


Figure 6.31 Hypothesis 2: The threshold for the terrain suitable for buildings is up to 12°. This scenario returns a total amount of 615 houses.



Figure 6.32 Hypothesis 3: The threshold for the terrain suitable for buildings is up to 13°. In this case, the total amount of houses is 765.



Figure 6.33 Hypothesis 4: The threshold for the terrain suitable for buildings is up to 14°, which allows 965 houses.

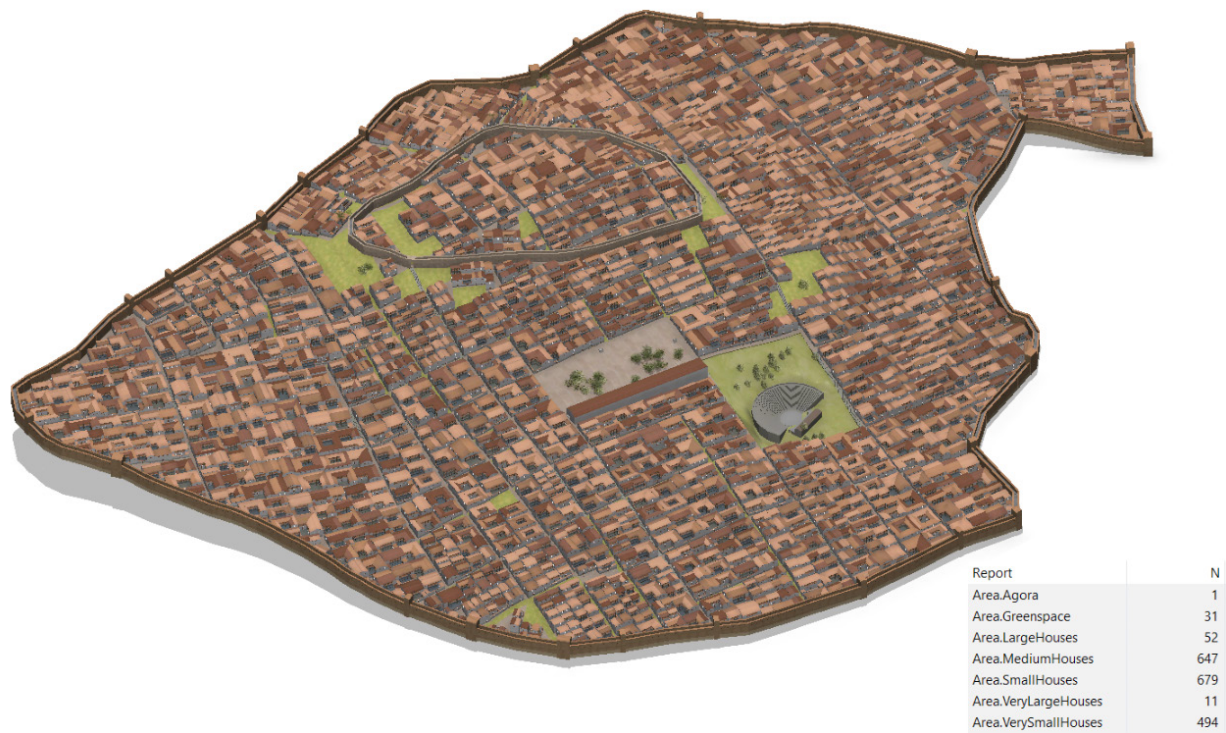


Figure 6.34 Hypothesis 5: The threshold for the terrain suitable for buildings is up to 24°, which allows 1883 houses.

more households could share the same house, the number of slaves is difficult to assess.<sup>1260</sup> By using these figures, about half of Koroneia's intramural space would have been occupied by houses, which would make up a total of ca. 510-561 houses corresponding to about 2,800 and 3,080 people (yellow line in Figure 4.32) and 555-610 houses corresponding to about 3,050 to 3,360 people (orange line in Figure 4.32).

At Koroneia, *in situ* stretches of walls have been recorded up to around 24° of gradient. The evidence we have, however, is too limited to establish whether these walls (sometimes just 3 or 4 adjacent blocks) used to belong to houses or more ephemeral types of buildings. It is possible that the continuation of the survey and the integration of geophysical results with the pottery data will lead to more refined conclusions on the use of the hill in antiquity, which are not possible to draw at this stage of research. Modern construction guidelines and comparison from archaeological sites differ in relation to the maximum slope suitable for construction. In modern guidelines, in fact, it is stated that most building can occur easily without major regrading on slopes between 0 and 10% (0 - 5.71°), moderate grading may be needed to support development on slopes 10 - 15% (5.71° - 8.53°), more care with grading is needed so that extremely steep slopes are not created by regrading on slopes 15 - 25% (8.53° - 14.04°), and extreme care should be taken when building on slopes of 25% and greater so slopes that are even steeper than 25% are not created.<sup>1261</sup> Although only a few papers report clearly the maximum slope that was used for construction at archaeological sites, when this piece of information is present it is clear that in many cases hilly terrains were exploited for construction up to very steep slopes. At Ephesus, for example, Groh reports that the preferred slope was up to 30° of gradient,<sup>1262</sup> while at Petres of Florinas, although the maximum gradient for construction on slope is not given, the overall hill's gradient was between 30 and 45% (16.70° - 24.23°).<sup>1263</sup>

For our purposes, I have created alternative reconstruction hypotheses taking as threshold a slope of 9°, 12°, 13°, 14°, and 24° (maximum slope of *in situ* blocks recorded at Koroneia). The insertion in the rule file of the possibility to automatically report on entities' count and areas makes it easy to calculate the number of houses that were allowed within the given space and the areas that domestic architecture would occupy in percentage. The results are shown in figs. 6.30 - 6.34.

Calculating the number of houses allowed within the city walls gives a starting point for the formulation of population figures for 4th century Koroneia. Our method, in comparison to the approaches briefly mentioned above, allows for the iterative formulation of different hypotheses, by permitting us to change both the parameters of house sizes and the slope's critical threshold, so as to visualize different scenarios, and maintaining at the same time an intellectually transparent approach when it comes to clearly establish the parameters for the calculations. The simulation which takes 9° as the critical slope value generates 321 houses, which would house about 1,760 people if we take the minimum average of 5.5 people per household suggested by Hansen. This value, however, leaves large parts of the town unbuilt where the survey has instead indicated the presence of houses. Despite the fact that currently these walls cannot be dated with certainty, as we lack contextual information (e.g. stratigraphical control), we can assume that this threshold is a too conservative value, resulting in the limited exploitation of the suitable terrain for building.

<sup>1260</sup> Census data for Hellenistic and Roman Egypt depict in fact a complex picture of the inhabitants of houses, see Hansen 2008, 259-86, esp. 276-80, citing Clarysse and Thompson 2006.

<sup>1261</sup> Torti Gallas and Partners-CHK, Inc. and Dodson Associates, 'The Neighborhood Model: Building Blocks for the Development Areas' (March 22, 2000), 106.

<sup>1262</sup> Groh 2012, 68.

<sup>1263</sup> Adam Veleni 2000, 35.

Also the value of 12° degrees leaves unbuilt a part of the town in the north-eastern sector where some structures were identified during the geophysical survey. At the other end of the spectrum we considered, if we take the value of 24° as threshold for building construction and settle for the minimum average of 5.5 people per household, the 1,883 houses allowed within the walled town would have housed more than 10,000 people, which, despite the problems related with the use of army figures as discussed above, seems to be too high a number for a medium sized town such as Koroneia – even more so since the army figures include also the population of Koroneia's *chora*, while our calculation takes into account only the people inside the city walls. The numbers of houses estimated by using slope values above 12° and below 14°, which returns about 3,380 and 5,300 people respectively, correspond to the advised slope values in modern construction guidelines and result in numbers closer to the calculations based on historical sources. This leads us to suggest that the laying out of the town on the hill was made by attempting to find a good balance between maximizing the exploitation of the available terrain for construction and minimizing the effort needed for building on a slope.

Given the fact that the rules report the number of houses by size, our method would allow us to set a different number of inhabitants per house in relation to their dimensions. We could suppose for example that smaller households would have had fewer slaves, while larger houses could house more people (we know that size should not be taken as an indication for wealth, but it is likely that a large house could be shared by more households and allowed lodgers as well), and construct a more refined model of population figures at Koroneia. Obviously we cannot claim that with our method we have reached more accurate figures than with the previously mentioned approaches, but we believe that we have created a model which is explicit in its assumptions and can be interactively used to formulate site-specific hypotheses.

#### **6.3.4 Exporting procedurally modelled Koroneia**

As I have argued in the previous chapter, 3D reconstructions are traditionally associated with public outreach, both positively connoted as an approachable means of communication of otherwise difficult to comprehend archaeological evidence, but also, unfortunately still nowadays, negatively connoted as ways to show case technical advances in visualization techniques with little archaeological value. In our view, 3D reconstructions are a very well suited environment to reconcile rigorous research and communication. In the first stage of its development, in fact, the 3D visualisation is deployed as a tool to test different possible hypotheses and the initially schematic and conceptual 3D models can be used as a platform for discussion across the team of experts working on the site. When the 3D model (or a first draft) has passed the phase of revisions and discussion, it can be then used as a platform to communicate the archaeological hypotheses to a broader public.

The previous sections of the chapter focused in particular on the modelling strategy I have adopted for the creation of Koroneia's 3D reconstruction and on the analytical opportunities that this approach opens up, both within CityEngine itself with the possibility of creating different reconstruction hypotheses of the urban layout and within a 3D GIS, especially in relation to visibility analysis. This section is instead dedicated to the exploration of some of the currently existing possibilities to make the 3D reconstruction of Koroneia available to a larger public. As will be discussed below, the 3D reconstruction described in this chapter is currently available online. I have experimented moreover with the game engine Unity3D to set up a prototype navigation in the virtual environment and a VR experience using a cardboard viewer. This application could be further developed when the study of the surveyed material will be completed and the 3D reconstruction updated accordingly.



Figure 6.35 Koroneia's 3D visualization exported in the \*.3ws format and viewed in the CityEngine Web Viewer: a) Alternative scenarios can be compared using the swipe view; b) buildings can be tagged so that they are easily searchable, and as shown in c) the lighting conditions can be changed to see their impact on the 3D scene.

*Online publication*

There are many factors that have to be taken into account when searching for options to guarantee a wide accessibility for a 3D visualisation. Traditionally, 3D models that were available on the internet could be viewed only by downloading a plug-in (e.g. Cortona 3D viewer), which was not an ideal solution for a variety of reasons. The plug-in may in fact be incompatible with the operating system that runs on the computer, and users in public institutions usually do not have the administrator privileges that are required to install the plug-in. The awareness that this procedure precludes the accessibility of 3D content led to alternative solutions.<sup>1264</sup> Recently, the development of the Javascript API Web Graphics Library (WebGL) enables us to see and navigate through virtual scenes without having to download plug-ins. WebGL exploits in fact the graphic card of the computer and makes the 3D visualisation run directly on the hardware, which greatly speeds up navigation and rendering capabilities.

Based on WebGL technologies is the CityEngine Web Viewer, which is made available by ESRI and it is optimized for the handling of 3D scenes that are created in CityEngine (Figure 6.35).<sup>1265</sup> This web app was created to provide urban planners with a tool to be able to share and discuss their development scenarios with people that did not have CityEngine. During the modelling process in CE, the 3D models within the scene can be tagged and the various components of the scene can be organised in layers, and then exported directly to the Web Viewer, where it is possible to navigate through the previously set layers and swipe through different development scenarios, and to query the 3D visualisation in order to find specific buildings according to the tags that were assigned. This online platform allows moreover the user to interactively change lights in the scene, in order to see in real time the effect of sunlight and shadows on buildings and to add comments. These features have been exploited for Koroneia's 3D reconstruction: the two hypotheses displaying the two possible reconstructions of the northern part of the site can be inspected and compared using the swipe view (Figure 6.35, top); a selection of buildings has been tagged, which allows the user to search for and highlight them within the 3D scene (Figure 6.35, middle); different lighting conditions can be set to see the impact of sunlight on the town's built environment (Figure 6.35, bottom).

*Interactive navigation and Virtual Reality*

The scene created in CityEngine can then be used to create an interactive experience that allows the user to navigate through the virtual world, by relying on game engines. I have experimented with Unity3D,<sup>1266</sup> which offers the basic version of the software for free, it is able to apply the rules of physics, thus creating a realistic interaction with the virtual environment, and it is customisable by adding scripts in e.g. JavaScript and C#. Moreover, it permits the publication of the 3D content on several platforms (among others iOS, Android, Flash Player, Xbox360, Oculus Rift and Google

<sup>1264</sup> The European project CARARE identified Adobe PDF as the most widespread software, since it is installed in 85% of computers. Although the possibility of embedding 3D models into PDF's is not widely known, the technical support exists for years now and this format seems to be by far the best choice for exchanging and publishing 3D models in a stable and widely accessible form (see CARARE' s deliverable: Pletinckx and Haskiya 2011). 3D models embedded into a PDF can be rotated, zoomed in and measured. Moreover, it makes it possible to change the light and its direction, in order to appreciate features that may be visible only under certain lighting conditions. Also, one can link the text of the PDF to parts of the 3D model, so that when the text is activated by a mouse click, the 3D model rotates to show the part that was linked to the text. Examples of 3DPDF's can be found at <http://pro.carare.eu/doku.php?id=training-materials:3d> (last accessed March 2017).

<sup>1265</sup> The 3D scene can be accessed via this page: <https://www.universiteitleiden.nl/en/staffmembers/chiera-piccoli#tab-1>; whether or not WebGL is supported/enabled in the browser and instructions to enable it can be found on <https://get.webgl.org/> (last accessed March 2017).

<sup>1266</sup> <http://unity3d.com/unity/>

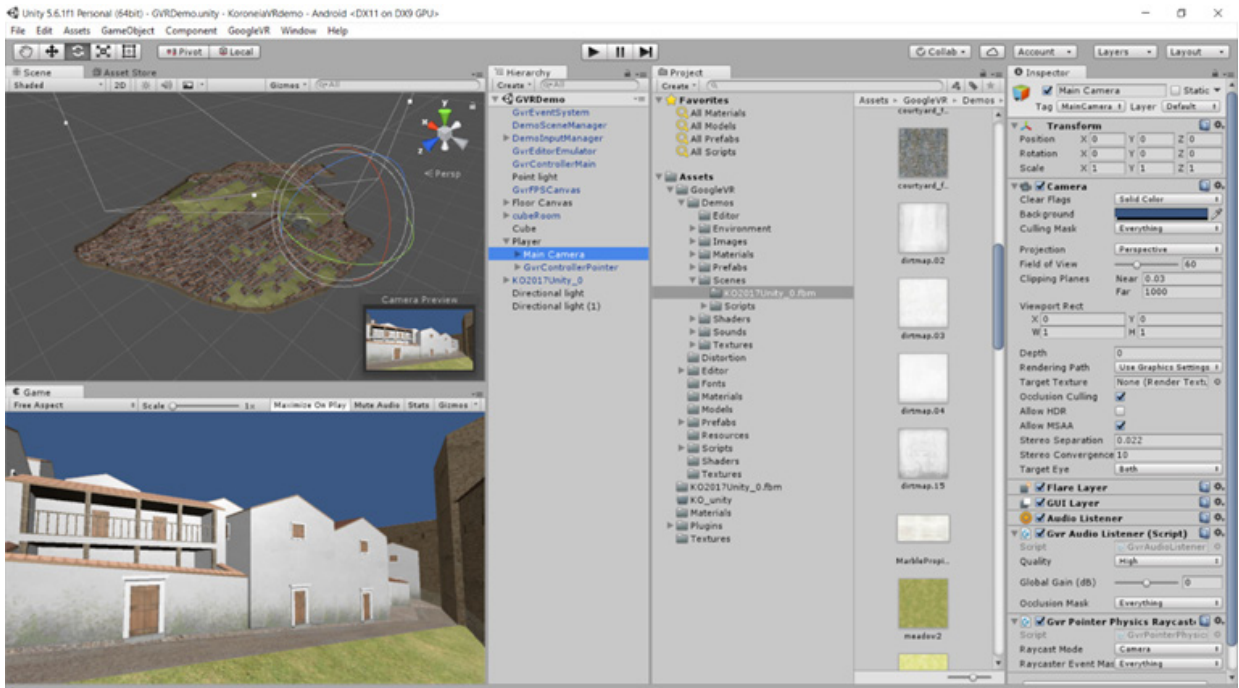


Figure 6.36 Screenshot from Unity3D showing the 3D reconstruction of Koroneia both from a bird-eye view and as it would appear as a first person navigation.

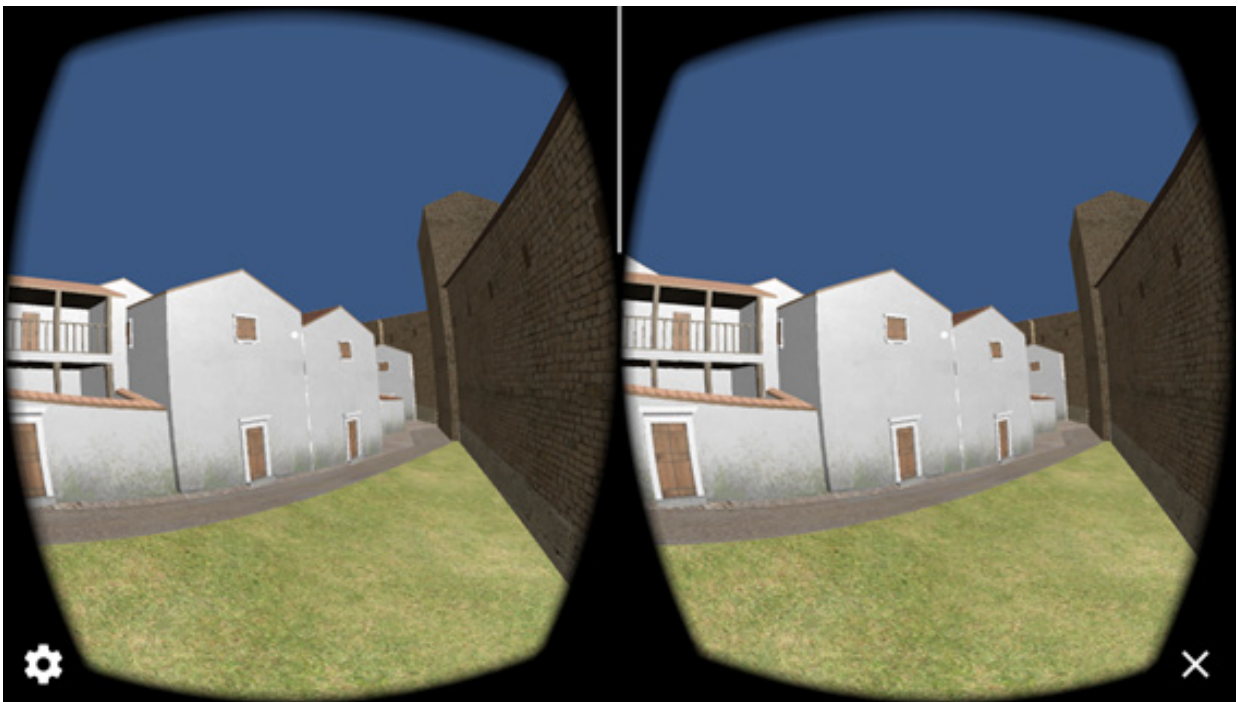


Figure 6.37 A screenshot of the VR experience with a smartphone and a cardboard VR viewer.

Cardboard) in a quite straightforward way, thus enabling an easy re-purpose of the 3D visualisation. The import of the scene created in CityEngine to Unity3D is straightforward using the Autodesk proprietary format FBX, which maintains geometry and textures with no loss of information.



I implemented a walk through the virtual environment of Koroneia, based on a first person controller navigation which allows the user to experience the urban fabric from a pedestrian's point of view (Figure 6.36). From Unity, it was possible moreover to create a prototype VR experience by using a smartphone and an inexpensive cardboard VR viewer which allows the user to stand in a location of the virtual environment and look around (Figure 6.37). Apart from the goal of engaging a public of non-specialists in the exploration of archaeological evidence, from a cognitive point of view, the navigation in the virtual world is useful to enable observations or raise questions that may have not been triggered by merely looking at the picture displaying the final rendering of a 3D visualisation. Observations on how the space is experienced and perceived can be made during the exploration of the virtual environment from within. It is possible to see, for example, what is visible from a certain place and how a change in lighting conditions affects the perception of space. In the case of Koroneia, for example, walking through the virtual environment allows the user to experience how narrow some of the streets were and shows the visual connection between focal points in the urban layout, such as the agora and the acropolis, from the pedestrian point of view.

For the time being, this prototype is used for research purposes only. In the future, however, when the processing of the survey data will be completed, this application can be further developed for heritage purposes. Notwithstanding the wealth of pottery sherds scattered across the site, an occasional visitor or an inhabitant of the nearby villages would struggle to understand the archaeological and cultural value of these apparently unstructured archaeological remains. This fact has unfortunately a major repercussion on the preservation of this ancient city that, as for nearly 80% of Greek sites, is in poor condition and is at risk to disappear. An app that allows the exploration of Koroneia's reconstructed 3D environment could be created as an educational tool, with the involvement of local communities.<sup>1267</sup>

It must be noted, however, the development of such a virtual tour is far from trivial. In fact, a great deal of criticism has been directed at virtual heritage applications that have been developed in the last years, concerning the lack of meaningful content, engagement, and the sense of place, and also the use of expensive technologies that require additional high costs and specialized personnel to guarantee their long term maintenance.<sup>1268</sup> Many factors play a role in how to devise such an interactive tour in order to make it a pleasant and learning experience. The definition of the target group of users is one of the most important aspects to calibrate the storytelling, the type of information that can be accessed and the type of interface that users can find most comfortable to interact with. Ideally, therefore, the application development should include an evaluation phase to monitor the user experience to assess usability, engagement and effectiveness in terms of knowledge acquired, to be able to address any issues.<sup>1269</sup> The time and effort spent for the development of such a virtual visit should also ensure the long term accessibility and maintenance of the applications, in order to avoid that an update in the operating system of the device causes the application to be incompatible with the new OS, thus rendering it useless. These elements will have to be taken into consideration in case of a future implementation of the above mentioned app.

<sup>1267</sup> See Bintliff 2013b.

<sup>1268</sup> A comprehensive review of critical remarks about virtual heritage is given in Tan and Rahaman 2009; see also Pujol and Economou 2009.

<sup>1269</sup> Gockel *et al.* 2013.

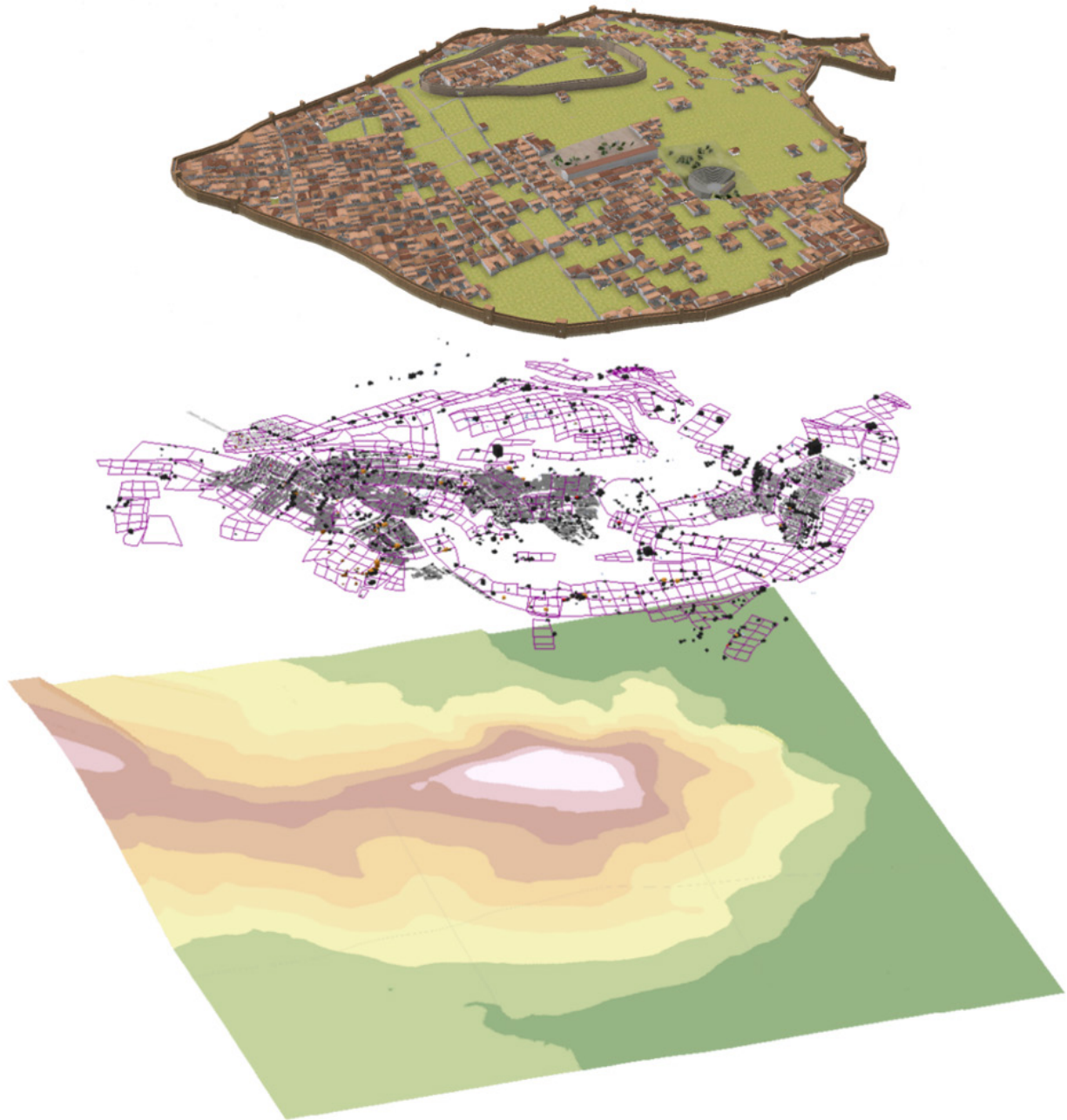


Figure 6.38 The main layers that compose Koroneia's 3D GIS environment. Bottom: terrain layer; middle: survey data (for the moment limited to architecture and geophysical prospections) and survey grids; top: one of Koroneia's procedurally modelled reconstruction hypotheses based on the current state of the data.

## 6.4. Discussion

### 6.4.1. Koroneia's 3D reconstruction: Intellectual transparency and reusability

One of the main aims of this study was to develop Koroneia's 3D visualisation taking on board and implementing the London Charter and Seville principles guidelines. As the developers of CE have argued in one of their first papers dealing with the application of this software to historical sites, a procedural

modelling approach allows the creation of intellectually transparent and London Charter-compliant 3D models.<sup>1270</sup> In fact, as previously demonstrated, the use of procedural modelling techniques can help us to create multiple hypotheses in a more time efficient way that can then be presented as alternatives, thus conveying the degree of certainty of each element in the reconstruction. The features that are archaeologically documented would remain in fact constant in all the 3D models, while the uncertain parts would be displayed differently according to the range of possibilities that the archaeologists consider as plausible. Moreover, the presentation of multiple reconstructions allows the 3D modeller to avoid choosing a 'peremptory single reconstruction', which is, as already mentioned, one of the problematic aspects of 3D visualisations in archaeology.<sup>1271</sup> While the creation of a 3D model for each possible hypothesis is a very time consuming task in a CG software (and for this reason this solution is rarely adopted), procedural modelling allows the display of multiple reconstructions by changing some parameters' values, as previously shown in the case of Koroneia.

Moreover, procedural modelling allows the insertion of comments in between the lines of the rule files that can, in this way, be enriched with references to all the sources that were used as the basis for the reconstruction and with comments regarding the modelling choices that have been made. The CGA shape grammar being a scripting language, such information can be marked as comments by using a double slash (//) or a hashtag (#) in case of one line comments, or by opening a multi-line comment with /\* and closing it with \*/ (as shown in Figures 6.13, 6.23 and 6.24). This type of information was included in the rule files for Koroneia, which turned out to be useful for two reasons. On the one hand, these comments provide a quick reference to remember the choices made in the modelling process and the data used as comparison. On the other hand, this material allows the rule files to become self-standing sources of information that can be shared as such. In this way, it becomes more accessible and understandable how the 3D reconstruction was developed and according to which data or comparisons the model came into being, which is more difficult to achieve when modelling in a CG software.

Also the exploitation of the interoperability between CityEngine and ArcGIS was geared towards the creation of an intellectually transparent visualization. Since the procedurally modelled 3D reconstruction is created on top of the georeferenced terrain, the former shares the same coordinate system as the original data. In doing so, it can be exported to ArcScene to check the correspondence between the reconstruction and the GIS layers (Figures 6.28 and 6.38).

Besides providing the tools to create an intellectually transparent 3D visualization, procedural modelling also allows an easier re-usability of 3D models. This topic is of particular importance as re-usability goes hand in hand with sustainability and cost/time effectiveness. One of the criticisms raised against archaeological 3D models is in fact that they are quite time consuming to make, and that they are rarely re-used in other projects. With procedural modelling, rule files can be easily shared, and lines of the scripts can be re-used in other rules. It must be noted, however, that the CGA shape grammar can be interpreted by CityEngine only, which limits the reusability of the rule files to this specific software. Nevertheless, all the variations that are procedurally generated can be exported (for example in OBJ or Collada DAE formats), thus allowing wide compatibility with other software packages. In this way, libraries of objects can be easily created that can be combined to create more complex virtual representations. Niccolucci has sustained that, for this reason, 'this methodology will ultimately free users from the need to use skilled craftsmen laboriously to build the 3D models from scratch'.<sup>1272</sup>

<sup>1270</sup> Haegler *et al.* 2009; see chapter 3, 58.

<sup>1271</sup> Forte 2000.

<sup>1272</sup> Niccolucci 2012, 29.

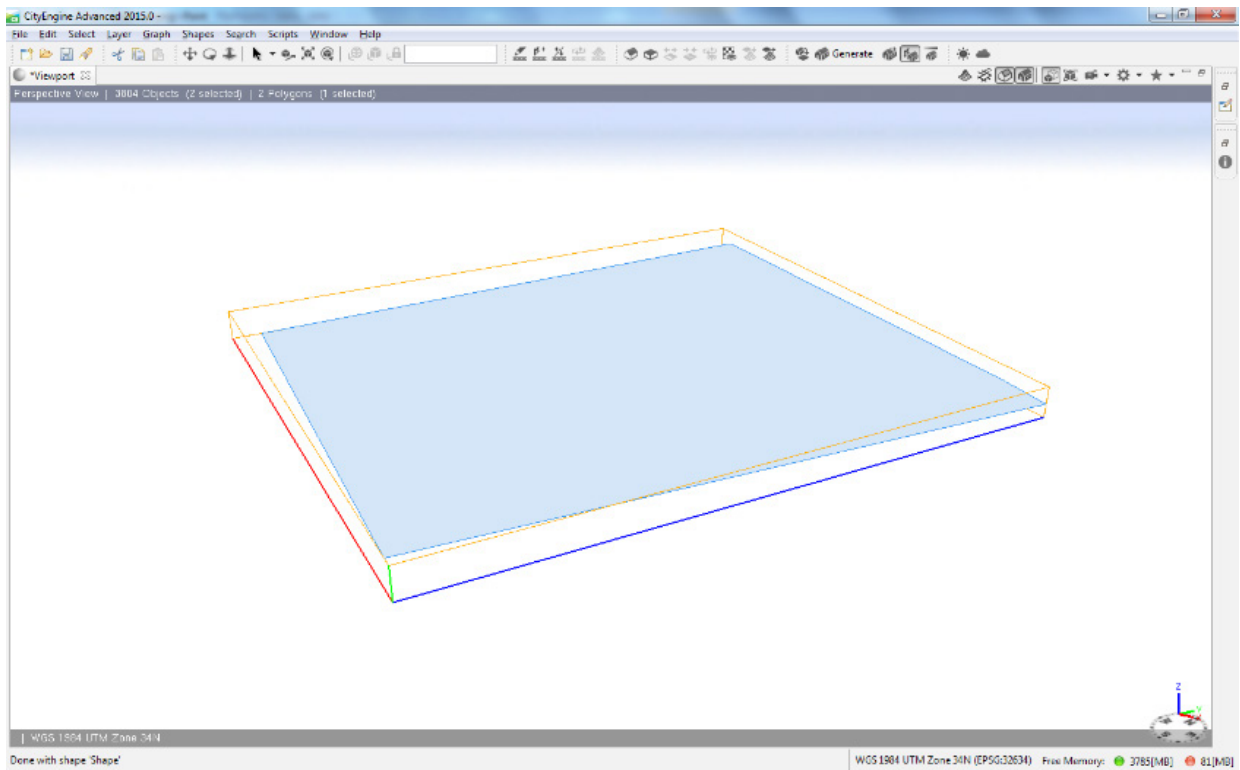


Figure 6.39 A screenshot from CityEngine showing the reference system of the initial lot, which consists of an oriented bounding box ('scope') governing all the operations that are performed on the shape.

#### 6.4.2. Rule-based modelling: challenges and work-arounds

The approach based on procedural modelling, although not devoid of complexity and challenges, proved to be suitable for the display of alternative reconstruction hypotheses of Koroneia in a (semi) automatic way. Moreover, the chosen software CityEngine provides a seamless interaction between the procedurally modelled scene and GIS, as GIS data can be directly imported in CE maintaining their spatial information and their attributes, and CE 3D scenes can be exported to ArcScene in gdb format. In this way, analysis on the built environment is enabled and original data and reconstruction hypotheses overlap, which fosters an intellectually transparent approach to 3D visualizations, further developed by the rule based modelling approach. Besides GIS, this software interfaces well with other environments such as e-on VUE (or Pixar RenderMan) for realistic renderings and Unity3D for interactive virtual tours. The methodology that we have adopted allows us to use the 3D environment as a tool to work with, in order to discover associations between data and evaluate different hypotheses, in line with Gillings' statement that opens this chapter.

Despite the high level of customization that can be reached with the CityEngine's CGA grammar, I have encountered some limitations that required *ad hoc* work-arounds when modelling an archaeologically correct ancient Greek architecture. One of the earliest obstacles I have faced during the creation of this rule set was setting the constraints for the orientation of buildings that could be meaningful in antiquity and the consistent indexing of each shape's axes. In our case, I wanted the houses' courtyards to face a constant direction (towards south or east), as these are the preferable orientations to ensure the best sun exposure. CityEngine's shape grammar includes the possibility to align each shape's scope to the world coordinates, thus allowing control of the geometry's generation. However, in complex geometries this measure reveals shortcomings in the consistent treatment of the scope's alignment across the numerous instances that populate Koroneia's urban environment.

```

y_angle = convert(y, pivot, world, orient,0,0,0)

setpivot_index_uncorrected = floor((y_angle+45)/90)
setpivot_index0 = setpivot_index_uncorrected+0 - floor((setpivot_index_uncorrected+0)/4)*4
setpivot_index1 = setpivot_index_uncorrected+1 - floor((setpivot_index_uncorrected+1)/4)*4
setpivot_index2 = setpivot_index_uncorrected+2 - floor((setpivot_index_uncorrected+2)/4)*4
setpivot_index3 = setpivot_index_uncorrected+3 - floor((setpivot_index_uncorrected+3)/4)*4

//slopeDegrees(sx,sy,sz) = atan (sy / ( (sx + sz)/2 )) This function was used in an earlier
version, CE has now an in-built function to calculate the slope:
slopeDegrees(sx,sy,sz) = geometry.angle(maxSlope)

@StartRule
Lot --> alignScopeToAxes (y)
    //print (slopeDegrees)
    Lot1

Lot1 -->
    case slopeDegrees(scope.sx,scope.sy,scope.sz) > 13 :
        report("Area.Greenspace",geometry.area)
        LotVegetation

    else:
        alignScopeToAxes(y)
        innerRectangle(edge)
        {shape:
            t(0,-2,0)
            extrude(world.up,2)
            comp(f)
            {top:
                alignScopeToGeometry(yUp,0)
                alignScopeToAxes(y)
                setPivot(xyz, setpivot_index1)
                // in this case I chose the N-E corner of the shape's scope
                as target pivot
                BuildingArea
            |side:
                alignScopeToGeometry(yUp,0)
                TextureFoundation
            }
            |remainder: TextureFoundation
        }
}

```

Figure 6.40 Lines of the CGA rule for domestic architecture which allows a more robust control on the reference system of the initial lot.

To better understand this problem, some clarifications are needed on how CityEngine handles the shape generation. Each shape that is created by the software (from lots to geometry created via procedural rules) has a set of locally interrelated coordinate systems – the object, pivot and scope coordinate systems – upon which all the geometric operations are based (e.g. rotation, translation, scale, split and component split).<sup>1273</sup> According to the CityEngine's guidelines, in the object coordinate system, the origin is placed at the first point of the initial shape's *first edge* (which is identified as the edge facing a street and therefore the 'front'), and the axes are oriented such that the x-axis is along the first edge, the y-axis is along the first face's normal and the z-axis is perpendicular to the former two.<sup>1274</sup> The pivot is located at the origin of the shape and when a component split is performed, it is placed at the first point of the first edge of the component, with x along the first edge of the face and z along the face's normal. The scope coordinate system relates to the pivot's orientation and it is composed by an oriented bounding box, specifying the length and orientation of its x, y and z axes (Figure 6.39). When a split or a component split (which allows one to divide a shape

<sup>1273</sup> See CityEngine's help for further information ([http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/is/create/block\\_params.html](http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/is/create/block_params.html))

<sup>1274</sup> See CityEngine's help for further information ([http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/cga/shape\\_operations/cga\\_so\\_trans\\_coordinatesystems.html](http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/cga/shape_operations/cga_so_trans_coordinatesystems.html))

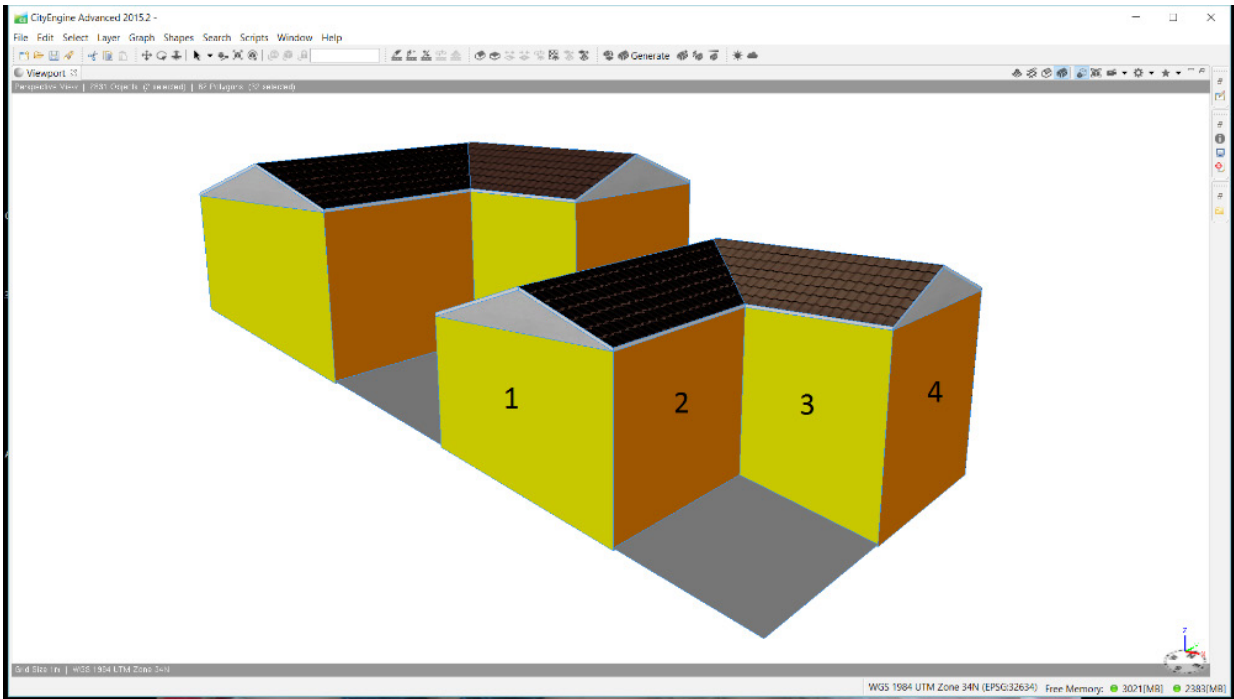


Figure 6.41 The problematic treatment of building facades in default L-shapes: using CityEngine’s standard syntax faces 1 and 3 cannot be distinguished and neither can 2 and 4, which makes it challenging to assign a different texture to facades facing the courtyard with respect to those that face the street. The adopted solution is given below.

```
LshapedHouseFacades_i(dx, y1, dy, dz, sx, sz) =
    LshapedHouseFacades_i2((sin(y1)*dx+cos(y1)*dz)/sx, 2+ (dy+45)/90, -(cos(y1)*dx-
    sin(y1)*dz)/sz)

LshapedHouseFacades_i2(xt, yt, zt) =
    LshapedHouseFacades_i3(1*float((xt>0.1)&&(xt<=0.9)) + 2*float(xt>0.9), floor(yt) -
    floor(yt/4)*4, 1*float((zt>0.1)&&(zt<=0.9)) + 2*float(zt>0.9))

LshapedHouseFacades_i3(xt, yt, zt) =
    3*float((yt==0)&&(zt==1)) + // east side, court
    4*float((yt==3)&&(xt==1)) + // south side, court
    5*float((yt==0)&&(zt==0)) + // east side, street right
    6*float((yt==1)&&(xt==0)) + // north side, street
    7*float((yt==2)) + // west side, street
    8*float((yt==3)&&(xt==2)) // south side, street
```

Figure 6.42 Lines of rule that allow a consistent indexing of the L-shaped building’s facades.

into its geometric components, such as faces, edges and vertices) is performed, a new coordinate system has to be assigned to the newly created children shapes, which usually take on the coordinate system of the predecessor.

In our case, the problem lies in the identification of the first edge of the initial shape (which is then passed on to the shape’s successors) as the edge facing the street. Our lots are randomly distributed within each block and therefore their first edge will not have a consistent position across the various shapes. However, such consistent orientation was needed to create houses that had their courtyards facing a specific, pre-determined direction, in such a way that the appearance of each procedurally modelled facade and roof could be controlled. CityEngine allows the user to manually select the first edge, however this procedure

would have been unfeasible for the high number of shapes that needed to be manually checked. In fact, Dobraja specifies that in the case of 1795 buildings that composed her 3D scene of Bavarian rural buildings, it took about 5 hours to manually change the first edge of each shape to the desired location.<sup>1275</sup>

As I wanted to ensure the maximum automatization of the town creation – especially given that the scene would be expected to be remodelled several times due to the ongoing nature of the survey – I have included a different approach within the rules that allows a more robust treatment of the pivot of the initial shape. Instead of taking the street position as reference point, a series of mathematical functions compares the current pivot's position with that of the target pivot (for example, the farthest NE corner of the shape to obtain a courtyard oriented towards S or SE) and determines how many steps the pivot has to shift to create the same angle with the world axes as the target pivot. This work-around concatenates several attributes whose values are calculated on the fly and are assigned as input value to the next one (the corresponding lines are shown in Figure 6.40).<sup>1276</sup> Specifically, the attribute *y\_angle* calculates the orientation of the y-axis of the shape's pivot in relation to the world axes, using the *convert* function. Next, the attribute *setpivot\_index\_uncorrected* uses the *floor* function<sup>1277</sup> to assign the input value of *y\_angle* to a quadrant (i.e. if *y\_angle* is comprised between  $-45^\circ$  and  $+45^\circ$ , then *setpivot\_index\_uncorrected* = 0; if *y\_angle* is comprised between  $45^\circ$  and  $135^\circ$  then *setpivot\_index\_uncorrected* = 1, etc.). Finally, a set of attributes *setpivot\_indexN* determine the amount of steps that have to be taken to arrive at the target pivot from the initial pivot. Since the situation can occur that the results give values that are outside the 0-3 range (e.g. -1, -2 or 4, 5, etc. in case the angle is calculated cyclically), multiples of 4 have to be subtracted to obtain a number within the 0-3 range as a final result. Each of the *setpivot\_indexN* attributes can then be used as parameter of the *setPivot* function when control is needed on the orientation of the shapes' pivot, assuming that the shape has 4 vertices. Specifically, in our case, these attributes have been crucial to select the starting pivot that guides the creation of the roof shade so as to guarantee the correct orientation of the slanted roof towards the courtyard.

Another challenging aspect of modelling using the CGA grammar instead of a manual modelling approach regarded the creation of a roofed court. The majority of the other archaeological projects that have used CityEngine have focused on Roman architecture. In a typical Roman house, the courtyard is surrounded by a peristyle, which can be modelled easily in CityEngine by generating an offset to an initial rectangular or square lot, extruding the resulting shape to the desired height, and assigning a sloping roof to the top faces.<sup>1278</sup> In our case, in contrast, the majority of the houses had a roofed porch or a wing around some, but not all, of the courtyard's sides. As a result, each of the facades had to be treated in a different way. The standard functions of CityEngine, however, did not give enough control over individual facades after performing component splits, especially when using the built-in *shapeL* and *shapeU* (see Figure 6.41), which is why a function was needed that consistently indexed the facades of a building. Briefly, a hierarchy of functions was developed that calculated the coordinates and orientation of each facade relative to the original shape's bounding box, and then converted these values into an integer range (1 to 8 for U-shaped buildings; 3 to 8 for L-shaped buildings) indicating each individual facade in a consistent way. For more detail, see Figure 6.42 for the rules for indexing the facades of L-shaped buildings.<sup>1279</sup>

<sup>1275</sup> Dobraja 2015, 31.

<sup>1276</sup> I thank J. B. van Klinken for his suggestions on how to deal with this aspect, and for his advice on using the *floor* function to evaluate and convert the pivot position to the target orientation.

<sup>1277</sup> The *floor* function gives the *greatest* integer less than or equal to *x* (e.g. the floor of 2.4 is 2). The opposite to the *floor* function is the *ceiling* function which returns the *smallest* integer that is greater than or equal to *x* (in the example, the ceiling of 2.4 is 3).

<sup>1278</sup> A similar shape can be obtained with the CityEngine in-built *OShape*, but due to how this in-built shape handles the top face, it would not be possible to create the inward sloping roof typical of peristyles.

<sup>1279</sup> I thank J.B. van Klinken for suggesting to me this way to formalize the indexing of the buildings' facades consistently.

Finally, a challenging issue was to create houses with open courtyards and/or open porches that are situated on a hilly terrain. The easiest solution to ensure that the initial slanted 2D shape would not protrude through the courtyard was to create a foundation beneath the house that would elevate the house to the highest point on the lot. This solution, however, gave unrealistically high foundations for houses on steep slopes, so I chose to compensate the slope only partly by introducing a foundation. For this reason, I used a component split that allowed me to transform this initial 2D shape in the groundfloor of the building (by extrusion). For steep terrains the problem remained that the terrain would intersect the courtyard facades on the ground floor, leaving holes in the facades' geometry. In this case, I had to split the courtyard facades along the y axes and subsequently re-scale the upper part that was not intersected by the terrain to the full extent of the ground floor. For details, see the function **PorchGroundFloor1** in the rule file.

### 6.5. Conclusions and future work

This chapter presented the methodology aiming to automatize both the organization of Koroneia's survey data in a 3D GIS and the 3D visualization of several reconstruction hypotheses of one of the phases of occupation of the hill, corresponding to a moment of possible destruction and rebuilding of the town centre, and limited to the space within the city wall circuit of that period. By writing an *ad hoc* procedural rule, which enables a customization of the visualization that goes beyond the capability of ArcGIS attribute-driven and 3D symbology, I have created a visualization that provides an intuitive insight into the types of blocks that are present on the hill. These rules have been applied to the reconstructed terrain to obtain a series of possible reconstruction hypotheses for the city layout, which can be interactively displayed by changing the parametric values that are expressed in the rule files. To our knowledge, this is the first implementation of a set of procedural rules for (late) Classical Greek architecture.

The workflow implemented in CityEngine and ArcGIS was integrated by the use of Open Source and free software packages such as QuantumGIS for the quick creation of shapefiles of some specific find classes, GIMP for the creation and editing of the textures that I have used for the 3D models, Blender and SketchUp to manually model some of the 3D buildings that were inserted in the scene, and Unity3D for setting up the interactive navigation. In order to make the creation of different reconstruction hypotheses as automatic as possible, I have used attribute maps (obstacle, mapping and function) to easily switch between visualizations that are plausible given the state of the data. The reconstruction of the terrain and of alternative hypotheses for the possible town layout, besides being useful to visualize the uncertainty in our dataset, has allowed me to estimate population figures, by setting constraints (e.g. slope threshold, house's dimensions and land use) which are explicit and easily modifiable when needed.

The approach here presented embeds the 3D reconstruction within the cycle of hypothesis generation and creates opportunities to use 3D visualizations as interpretative and analytical tools at the service of the archaeological research. The approach is particularly suitable for a site, such as Koroneia, where the study and investigations are still in progress, which implies future updates of the resulting 3D visualization. This workflow allowed me therefore to focus on the *process* of 'reconstructing', and on the analytical aspects of such an endeavour, rather than on the end result. Moreover, the chosen procedural modelling approach allowed me to be explicit about my modelling choices and interpretations. This is an important point in any archaeological research and especially in the case of Koroneia where, as discussed in chapter 4, that there are still many open questions given the ongoing study of some finds categories.

The methodology here described entails the initial effort of compiling the rule files in a way that best suits the characteristics of the dataset and the scope of the project. In the long term, however, this



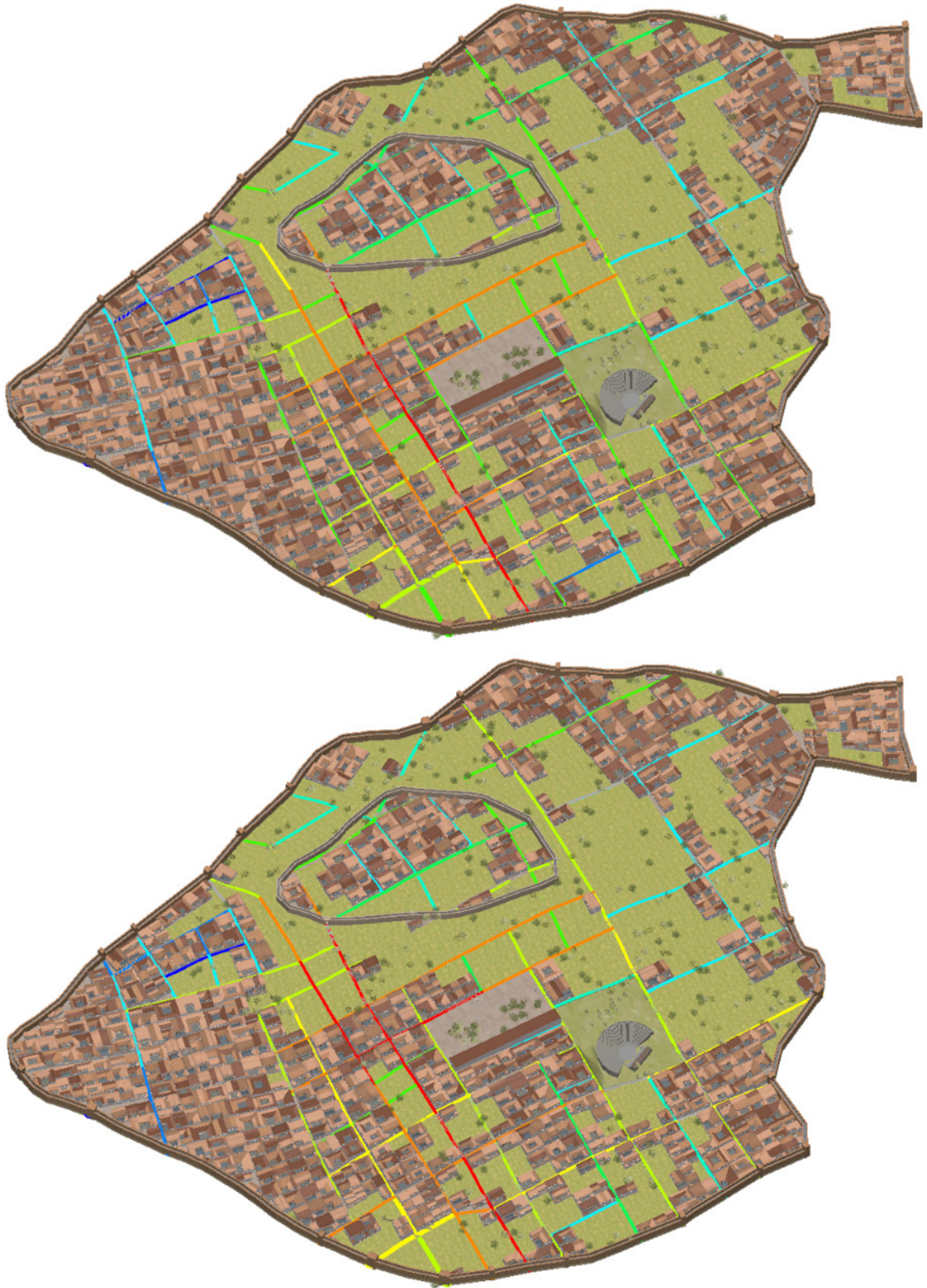


Figure 6.43 Results of the test analysis on global (top) and local (bottom) integration values of Koroneia's reconstructed street network.

approach results in a time efficient data handling, updating and displaying of alternative hypotheses which account for the different interpretations of the available data. This rule-based approach allows the recording of modelling steps to enable the assessment of the work by other scholars and fosters the possibilities of sharing and re-using existing assets for new projects, thus increasing the time and cost efficiency of the approach on the long term. Moreover, the focus on automation and iteration of the modelling process creates the premises to consider the 3D visualization as a tool to work with, instead of a mere representation of already acquired knowledge. To this end, the use of map layers associated with rule files as I show-cased in this chapter represents a time efficient workflow for the modelling of alternative reconstruction hypotheses and in view of the prospective updates needed in the long term.

The drawbacks of using this approach are related to the initial effort spent in learning CityEngine's shape grammar and to the technical aspects of the software installation, which, in its latest releases, requires powerful computers and graphic cards to be able to work.<sup>1280</sup> Moreover, the fact that this software is not backwards compatible and does not allow the user to save the scene in some previous version's format, as is the case in other software such as AutoCAD, heavily impacts the editing and sharing of the 3D scene in CityEngine proprietary format (\*.cej) with different users or on different computers. For a study that aims at intellectual transparency, the use of a commercial software for the development of the 3D visualization and of the CityEngine web viewer for its publication may be seen as problematic. However, CityEngine is currently the only software on the market which offers the integration of procedural modelling and GIS and thus fulfils the requirements of Koroneia's 3D visualization project. Moreover, the CityEngine web viewer is optimized for the online publication of 3D scenes that are created in CityEngine and allows the comparison of alternative reconstruction hypotheses by using the swipe view, as shown above.

Future work on this project should include the mapping in a 3D GIS of the settlements in Koroneia's territory (*chora*) and should also extend the 3D reconstruction to include the countryside, to create a holistic view of the functioning of the town within its broader context, as the surrounding territory has always been an integral part of the life of the town. Moreover, the same procedural modelling approach could be extended to other time periods, for example by creating or adapting the libraries of Roman architecture that have been already used in other projects that were mentioned in chapter 3. An additional map layer and related rule file can be created that colour codes the level of reliability of the reconstruction (e.g. green = based on survey data; yellow = inferred from survey data; orange = based on comparisons with other sites; red = conjectural).

Regarding the exploitation of the 3D reconstruction for analysis, there are more possibilities that can be explored – especially when the study of the survey data will be concluded and a more complete overview of Koroneia's urban layout can be offered. One future direction of research that I have started to explore is the analysis of the street network based on the principles of the Space Syntax theory, which relates the spatial configuration of the built environment to the social structure that created it.<sup>1281</sup> For the moment, I have run an analysis which computes the global and local integration values<sup>1282</sup> of Koroneia's street network by using the built-in 'analyse graph' tool in CityEngine. While global integration usually reproduces the patterns of centrality, thus giving predictable results, the local integration index is more suitable to highlight patterns of movements and areas of commercial activities at a local scale.<sup>1283</sup>

<sup>1280</sup> The system requirements for the 2016.1 release include 16 GB of RAM and OpenGL 4.1 or later (the complete list is available here <http://desktop.arcgis.com/en/cityengine/latest/get-started/esri-cityengine-system-requirements.htm>).

<sup>1281</sup> Hillier and Hanson 1984.

<sup>1282</sup> Global integration computes the closeness of an axial line in the street network to all the other axial lines, while local integration indicates closeness only up to a defined radius of topological distance.

<sup>1283</sup> Cutini 2009, 88; 95.

The analyses of Koroneia's street network show that the most integrated streets both at the global and local level are those bordering the agora on its eastern and western slopes (Figure 6.43). While at this stage it is necessary to be careful in advancing hypotheses based on the analysis of a fragmentary and partially reconstructed street network, it will be interesting to further develop this line of research. The area highlighted in the analysis corresponds indeed to the borders of what the survey has identified as the market place of the town, and to the core of the Roman settlement on the hill.

Besides the further development of a VR experience with a portable headset, another possibility that was discussed at the beginning of this project was the exploitation of Augmented Reality technology to create an on-site navigation where the reconstructed 3D buildings are superimposed on the real world. As already mentioned in chapter 3, AR is a popular technique and holds great promise for the future. The creation of outdoor, on site AR applications is however still challenging. So far, solutions have been offered by commercial companies that sell their development kits with different licensing options or promise to take care of all the process of creation of the AR application.<sup>1284</sup> The market of available platforms is however still fluctuating and the overall impression is that this technology still lacks standards and consistency, a *caveat* that was already voiced by a survey of the state of the art of this field in 2011.<sup>1285</sup> As a consequence, it is at present difficult to create applications that will guarantee to remain functioning in the long term and that work across several operating systems. Developments that are happening at this moment (especially regarding the release of Apple's ARKit and Google's ARCore development kit) however promise to change the situation and will certainly lead in the near future to an increased number of experimentations with this technology which will benefit also the heritage sector.

---

<sup>1284</sup> Current companies offering such services are e.g. Vuforia (<https://developer.vuforia.com/>) and Augment (<http://www.augment.com/>).

<sup>1285</sup> Butchart 2011.