

Parsing Architecture within Plan Drawings with Application to Medieval Castles and Fortresses

A. Willis^{1†}, Y. Sui¹ and K. Galor²

¹University of North Carolina at Charlotte, Dept. of Electrical and Computer Engineering, Charlotte, NC USA

²Brown University, Dept. of Judaic Studies, Providence, RI USA

Abstract

Plan drawings are graphical documents critical to the documentation of architectural features at historic sites. These drawings include important geometric information such as the location, shape, and size of architectural features, which, for decaying or collapsed structures, may be the only existing records of the intact structure. This paper discusses an algorithm that estimates the geometry and semantic interpretation of architectural structures from a plan drawing. The estimated values are used to automatically generate a 3D structure using the estimated semantic labels of structural elements in the plan drawing. We demonstrate the utility of this approach by parsing several plan drawings of medieval castles and fortresses and generating 3D reconstructions of these structures and detail typical circumstances that prevent the system from generating a valid reconstruction. Since the 3D model is derived from plan drawings where the architectural contour is well-defined, the approach automatically provides near-pixel level accuracy at all locations which is very difficult and time-consuming to guarantee when manually constructing 3D models from the same drawing. Hence, these automatically-produced models can provide unprecedented accuracy to the in-situ remains not feasible with conventional manual model-building techniques. While this article represents initial work on this topic with limited scope (castles/fortresses), we envision that subsequent enhancements to this method will be a valuable tool for efficiently generating accurate 3D models for many different historic structures.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

This article proposes prototype software for semi-automatic estimation of architectural geometry and semantic information in a plan drawing; a problem that we refer to as *architectural parsing*. Plan drawings are graphical documents that provide a top-down view of a site or a geographical region within a site. Plan drawings document the structure and spatial arrangement of objects within a geographic region and have been used extensively throughout written history for this purpose.

At large scales, these drawings can incorporate informa-

tion that indicates the structure of settlements or perhaps districts within settlements. Smaller scale plan drawings often indicate the structure of just one or two buildings and the spatial arrangement of rooms within these buildings. This article proposes a method for extracting architecture from these small-scale plan drawings and subsequently decomposing that architecture into a collection of geometric components and assigning semantic labels to one of more of these components. We cast the estimation as a parsing problem, typically encountered in computer languages and linguistics, where parsing algorithms extract tokens, e.g., words or syntax elements, to determine their functional or grammatical structure. Our parsing problem seeks to extract geometric tokens, i.e., pieces of architecture, and applies shape estimation methods to estimate the semantic labels for each token geometric. Heuristics, motivated by a hand-specified grammar, is then applied in a bottom-up clustering procedure that groups together tokens to estimate higher-order se-

[†] This work was supported in part by the National Science Foundation under IIS-0808718. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

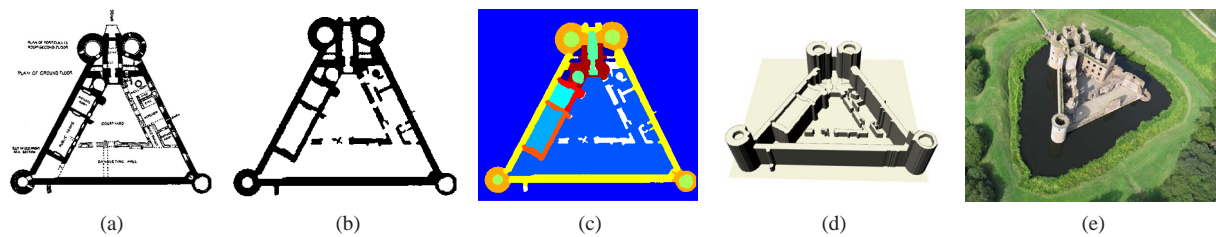


Figure 1: A summary of the proposed method for parsing an archaeological plan drawing. (a) original digitized image, (b) binary image with non-architectural features removed, (c) semantic labels are assigned to portions of the image, (d) a 3D model is estimated using the semantic labels, (e) an aerial view of the actual in-situ remains of the castle.

semantic labels that are constructed from groups of geometric tokens.

2. Related Work

This work relates to results from two somewhat distinct areas. Our model for estimating structures within archaeological plan drawings is related to ongoing research in document analysis where researchers are performing automated processing of architectural plan drawings. Our model for estimating semantic labels from recognized geometric tokens relates to shape modeling via shape grammars or, more generally, procedural modeling which has recently gained popularity within the graphics modeling community.

There have been several approaches provided in the document and image analysis literature that seek to parse objects within images. A generative approach for image parsing using Bayesian models is provided in [TCYZ03]. Here the authors focus on models that combine structure and appearance classify specific types of objects within the image; specifically faces and letters in an image. While this work is an important example of how Bayesian models apply to parsing problems, the models and training proposed are purpose built for faces and text letters in images and requires considerable generalization for parsing of generic images.

[DTASM00] is an early example of a method purpose built for processing architectural plan drawings. This method seeks to extract text information, various geometric information, and small-scale building substructures such as windows using a sequence of algorithms, each of which are detected and extracted using special purpose algorithms customized to each type of drawing annotation. Unfortunately, this method is geared towards contemporary architectural drawings that use standardized representations for structures such as windows, doors, stairways, interior / exterior walls etc. Archaeological structures often exhibit unique features or period-specific substructures, e.g., arrow-slits for a castle, which can vary widely in size and shape. Likewise, drawings of these structures do not have a standardized representation and the graphical representation of any given structure or sub-structure can vary significantly, even for different plan drawings of the same architecture.

[LMV01] presents a method for matching shapes modeled as a sequence of lines. Each of the regions enclosed by a collection of lines defines a region and lines are grouped by the regions that they bound. A region adjacency graph is then constructed where each region is a node in the graph and adjacent regions share an edge within the graph. Sub-graph matching techniques are then used to recognize instances of various shapes within a drawing. The method is applied to hand-specified drawings including hand-written architectural drawings. We also propose a graph-based model for the shape and topology of architectural structures. However, our graph is defined differently and allows for processing lines and open regions which are shortcomings of this approach.

[DL99] presents a method for vectorization of line drawings via a Space Pixel Vectorization (SPV) algorithm. The approach seeks to approximate the in terms of the medial axis of the lines present in the raster image. Much attention is given to proper preservation of junctions such as right-angle junctions. The method is applied for line drawings of mechanical parts. This approach fails to accurately preserve junctions between lines and arcs especially when the junction angle between these two contours is particularly small, i.e., when the two contours are nearly tangential.

[YWR09] provides informative survey on methods proposed for parsing contemporary architectural drawings. In addition to outlining several popular approaches for this problem the authors also demonstrate how these methods can integrate with procedural modeling techniques to generate 3D models of modern buildings. This work builds upon their previous work on procedural modeling for buildings [MWH*06] that could automatically generate synthetic model of realistic looking cities and undamaged, regular ancient structures such as a Mayan Puuc palace [MVW*06]. However, these earlier building models, while having believable exteriors, were not geometrically consistent with the actual buildings inside and out. Hence, such models are suitable for some visualization contexts but are not suitable for detailed archaeological analysis.

3. Motivation and Contribution

There are several important factors that motivate development of this system:

1. Many architectural structures no longer exist and plan drawing represent one of the best available sources for geometric information regarding these structures. Systems that respect the exact dimensions (or relative-dimensions) of the plan drawing provide 3D models that are highly accurate to the original source data.
2. Often plan drawings can contain errors and ambiguities, particularly when dealing with ancient documents. Models constructed from different plan drawings may be interactively compared and contrasted to provide new insights that would otherwise be difficult to gather. Especially when considering the third dimension, i.e., the height of structures depicted within the plan drawings.
3. Rigorous research requires careful consideration of all previous documentation. Much (almost all) of which is typically in analog, i.e., paper, form be it photographic, hand drawn, or the written word. Practitioners of digital archaeology must respect these old data sources while simultaneously recording new data using contemporary recording technologies and analysis tools. Software such as this can facilitate bridging the gap between new photographic and 3D scan data recorded from historically important architectural complexes and pre-existing plan drawings of the same building.

Factors (1) and (3) are of concern mainly to archaeologists and practitioners of digital archaeology while factor (2) is a problem generic to estimating 3D structure from plan drawings and applies to both contemporary plan drawings and those made in antiquity. The proposed software is a tool that address these issues by greatly expediting the time-consuming process of converting pre-existing analog pen-and-paper data into digital data that can be efficiently stored, transmitted, and analyzed. While our initial approach proposed in this paper is limited in scope, we envision that these tools will become indispensable for accurately digitizing architecture in cultural heritage.

Since the model is derived from plan drawings, accuracy of the 3D structure is comparable to that of the original drawing, i.e., our 3D model is an accurate geometric reproduction of the apparent contour of the structure as indicated in the plan drawing. Our approach automatically provides near-pixel level accuracy at all locations which is very difficult and time-consuming to guarantee when manually constructing 3D models from similar (or the same) drawings. Hence, these automatically-produced models can provide unprecedented accuracy to the *in-situ* remains not feasible with conventional manual model-building techniques.

4. Methodology

Our approach consists of seven steps:

1. Digitize the manuscript of interest with a digital camera or scanner.

2. Pre-process the image to remove text and line-drawing annotations.
3. Estimate a skeleton shape model for architectural elements present in the processed image.
4. Vectorize the estimated skeleton into simple, i.e., non-intersecting, curve fragments connected at a collection of special (x, y) locations denoted nodes.
5. Estimate a shape model for each of the curve fragments.
6. Using the estimated values from steps 1-5 and hand-specified heuristics, each pixel in the binary image is assigned a semantic label.
7. Based on the label for each curve fragment, we construct a 3D model over the extent of the curve fragment.

Steps 1-5 are generic to the problem of estimating architectural features from a plan drawing. However, steps 6 and 7 must utilize *a-priori* knowledge regarding the structure and geometric patterns typical to the architectural style being processed. For our examples, we incorporate *a-priori* knowledge regarding the structure and layout of a typical medieval castle.

4.1. Digitize the manuscript

The method used for the digitization of a plan drawing typically involves either (1) taking an aerial photograph of the document (non-contact) or (2) scanning the document. As in any analog-to-digital process noise is generated in the conversion and aliasing may occur especially for sharply varying structures and small-scale variations in the drawing. Geometrically accurate recordings of the printed document is the goal during the capture process and this topic has received considerable attention over the past two decades [HZ04] and we refer the reader to these references for details. For our experimentation, we used a flatbed document scanner set to a data capture resolution of 300 dpi (Figure 6(a)-top row), and we retrieved publicly available images from the web (Figures 6(a)-middle and bottom rows). As pointed out in § 4.7, the automatic parsing algorithm can generate incorrect results due to the high degree of variability present in plan diagrams (see Figure 2). We also assume that the plan drawing contains an indication of the enclosing castle walls, i.e., the castle walls form a closed contour within the image.

4.2. Pre-Processing the digitized manuscript

The goal of this step is to remove all information that does not relate to the architectural complex from the digitized drawing. We assume the drawing is a greyscale scan of a black and white drawing and proceed by estimating a binary image, I , from the digitized greyscale image using a simple threshold. Connected components, i.e., groups of contiguous 1s and 0s in the binary image are grouped together and assigned distinct region labels. The connected components computes a disjoint set of regions that covers the scanned image.

At this point we adopt mathematical notation to refer to

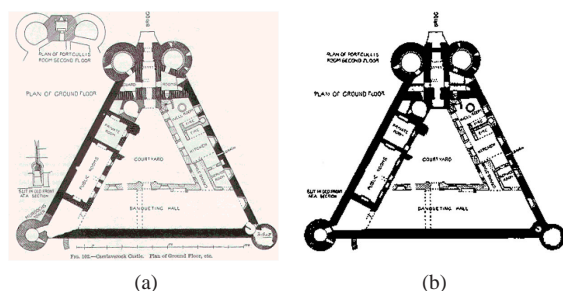


Figure 2: (a) An original plan drawing of Caerlaverock Castle. (b) A binary version of the image in (a). Note that some content has been manually removed before processing such as the inset gate detail in the top left corner and the front arrow-slit detail on the lower left side. Typically these modifications can be done quickly (< 5mins.) by filling-in or removing the problematic drawing components.

each of the connected image regions. Black regions and white regions are given distinct symbols. White image regions in plans typically denote an open space, i.e., locations not occupied by architectural building elements. For reasons that will become clear later, we refer to these regions as *faces*, and assign each such region a face index denoted F_i . Black image regions that remain after processing are assumed to be due to architectural features in the plan drawing and are the focus of processing for subsequent steps.

Architectural plan drawings often contain information that indicates the location small-scale architectural building elements within the larger complex. Common structures include staircases, windows, and doorways. However, for the purposes of estimating the general building shape, we remove these structures by replacing them with black pixels. Let $card(F_i)$ denote the number of pixels within the i^{th} white region. Our replacement procedure fills in all faces that occupy less than 0.05% of the total image are replaced by black pixels.

There is often a large amount of additional, i.e., non-structural, information contained in plan drawings. These often include line-drawing annotations, i.e., thin lines within the image, examples of these lines include topographical lines, geographical grid lines, excavation grid lines etc. These thin-line features in the plan drawing are removed by a sequence of open and close morphological operations using a 3x3 structure element S :

$$I_{new} = I \circ S \bullet S$$

where \circ denotes the open morphological operation and \bullet denotes the close morphological operation.

Apart from line-drawing annotations, there are often text annotations, and other small graphical markings, e.g., special features indicated by a legend, commonly included on plan drawings. These annotations are detected as small isolated black regions and removed using the same criterion as

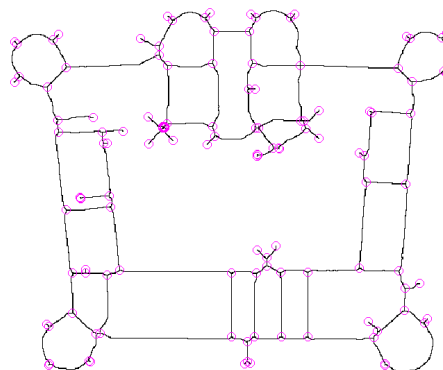


Figure 3: Skeletonization and vectorization

that used for the faces earlier, i.e., all connected black regions that occupy less than 0.05% of the total image are replaced with white pixels.

In summary, our processing of the image consists of five steps: (1) convert the image to a binary image; (2) group together contiguous sets of pixels having identical values in the binary image; (3) fill-in small white regions with black pixels; (4) perform morphological operations to remove lines in the image; (5) fill-in small isolated black regions with white pixels. Note that the order of this procedure is important given the non-linearity present in the morphological operations of step (4).

4.3. Extracting a skeleton from the processed image

The goal of this step is to convert each of the black regions associated with an architectural structure from a volumetric, i.e., pixel-based, representation to a curve, i.e., vector-based, form. Typical models applied for this are the edge thinning [LS92], skeleton computation [Soi03], and the medial axis [Ley08]. While researchers have indicated that the medial axis is a more accurate and stable representation, we opt to use the skeleton representation. This decision has both a pragmatic and theoretical motivation. Pragmatically, we wish to use each pixel within the plan drawings as a grid upon which we can place volumetric elements, i.e., building blocks. Hence, we only require assignment of labels to each pixel location. Additional accuracy afforded beyond this grid is wasted computation given this model. Theoretically, the skeleton provides us with a result that has the same form as the original data, i.e., the skeleton consists of a sequence of pixels in the image. Curve-based representations such as the medial axis require careful conversion between their continuous representation and the discrete manifestation of that shape model in the digital image. While these challenges have been tackled by researchers, we feel that the shape representation provided by a skeleton model is sufficient for this application due to: (a) the simplicity typical of large-scale ancient structures and (b) the algorithmic and computational complexity associated with using continuous shape models.

Our skeleton extraction process follows the method described in [Soi03] and uses a sequence of eight different morphological hit-or-miss operations that thin black image regions. The morphological elements are applied iteratively on the image until there are no changes in the image. This approach iteratively thins black image regions, i.e., regions indicating the presence of architectural structure, to a skeleton that is 4-connected almost anywhere (exceptions occur in black pixel regions similar to a disk).

4.4. Vectorizing the skeleton

This step converts the skeleton from a collection of (x, y) pixel locations to a sequence of curve fragments. Curve fragments are delimited at each end by *vertices* which correspond to special points on skeleton. Specifically, a node is located at each (x, y) pixel location where a skeletal curve ends or where three or more skeletal curves meet. Pairs of vertices serve as delimiters, i.e., end points, where skeletal curve fragments start and end. We then model the connectivity (topology) of the architectural structures using a graph model $G(V, E)$ where V denotes a set of vertices and E denotes a set of edges. For our graph, each vertex is an (x, y) vertex position extracted from the skeleton and denotes either: (a) a location where a wall ends or (b) a location where a wall junction occurs. Each of the skeletal curves extracted in the vectorization process corresponds to an edge in the graph and indicates the presence of a wall in-between these two vertices.

Let $V_i = \{\mathbf{p}\}$ denote the i^{th} vertex location within the computed graph containing the pixel location, $\mathbf{p} = (x, y)^t$, where the skeletal curve junction occurs. Let $E_j = \{(i_0, i_1) | \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{N_j}\}$ denote the skeletal curve fragment that joins the pair of vertices having indices (i_0, i_1) in the graph via a sequence of N_j pixel locations. It is important to note that multiple edges may exist between any pair of graph vertices which makes our graph representation somewhat different from those typically encountered in graph theory. Such situations tend to occur in drawings of highly symmetric structures which, due to noise in the drawing digitization and skeleton estimation processes, seem to occur rarely in practice. Our approach deals with these situations without need for special treatment. We express the vectorized skeleton in terms of the computed graph $G(V, E)$ where $V = \cup_i V_i$ and $E = \cup_j E_j$.

$G(V, E)$ is a *simply-connected planar graph* by virtue of the source data and our definitions for nodes and edges. Hence, we augment our graph with the definition of *faces*, i.e., regions bounded by edges, including the outer, infinitely-large region. Since edges correspond to skeletal curve fragments in the image, faces cover all of the open spaces in the image, i.e., every white pixel in the binary image will lie within some unique graph face. For notational purposes, let $F_k = \{\cup_j E_j | \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{N_k}\}$ denote the k^{th} region of white pixels bounded by the set of edges $\cup_j E_j$ and containing a set of N_k pixel locations that denote white pixels lying inside the region covered by the k^{th} face. Our

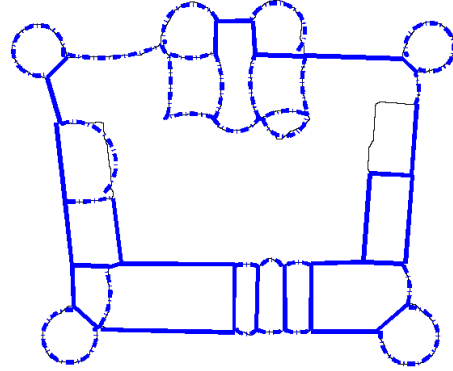


Figure 4: Fitting shape models to graph edges, i.e., skeletal curve fragments. Circular arcs are shown as dashed blue lines and linear segments are shown as solid blue lines. Black lines indicate points lying on the vectorized skeleton. Some edges have not been classified due to large fitting error for both of the fit shape models such as the room on the top left side (in black). Many leaf edges present in the original skeleton have also been pruned, as they do not affect the topology of the graph.

augmented graph is then $G(V, E, F)$ where $F = \cup_k F_k$ and each face contains references *only* to those white pixels lying within the region delimited by the face edges. The graph faces are computed using the classical wall-following algorithm which is capable of identifying all simple loops within graphs of this type.

4.5. Fitting Curve Fragment Shape Models

Using the computed graph model, we then estimate shape models for each graph edge, i.e., skeletal curve fragment. Towards this end, we fit a linear curve model and a circular arc curve model to the sequence of points. Fitting solutions are quickly computed using the method specified in [Tau91] which provides an explicit solution for fitting generic algebraic curves to 2D data. We then compute the Euclidean error between the fit model and the curve fragment data as indicated in equations (1) and (2).

$$\epsilon_{\text{circle}}(E_j) = \frac{1}{N_j} \sum_{j=1}^{N_j} \|\mathbf{p}_j - \mathbf{c}\| - r \quad (1)$$

$$\epsilon_{\text{line}}(E_j) = \frac{1}{N_j} \sum_{j=1}^{N_j} \frac{|\mathbf{p}_j \cdot (a, b)^t + c|}{\sqrt{a^2 + b^2}} \quad (2)$$

where \mathbf{c}, r from equation (1) denote the (x, y) location and radius, respectively, of a circle fit to the sequence of curve fragment points and (a, b, c) from equation (2) denote the coefficients of the fit algebraic line $f(x, y) = ax + by + c$. The parameters of the shape model having smallest fitting error are associated with each edge. These classifications are then

utilized by the procedural model to determine the style and type of masonry used to construct the wall.

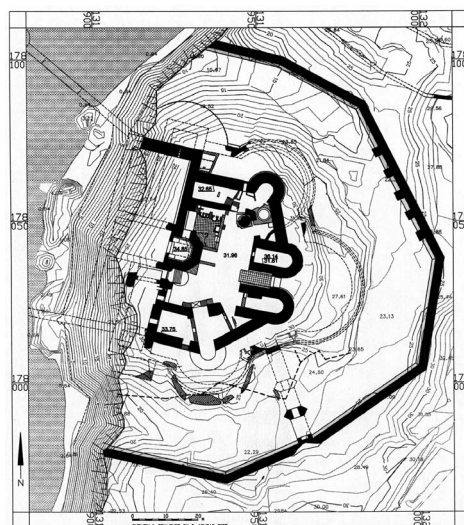
4.6. Estimating Semantic Labels

The final estimation step seeks to assign semantic labels to each pixel within the binary image. These labels are divided into two groups: face labels; semantic information associated with white pixel regions in the binary image and edge labels; semantic information associated with black pixel regions in the binary image. Specific labels associated with faces and edges depend largely on the type of architectural structures included within the drawing. Our approach concentrates on drawings of medieval castles and fortresses and uses a set of semantic labels appropriate for these structures.

Prior to estimation of the semantic labels, we simplify the computed graph by pruning, i.e., removing, *leaf edges*, i.e., edges that connect isolated vertices into the graph. These edges represent buttresses that stabilize and fortify the structure, extensions to the existing structure, or areas where a portion of the original wall has collapsed leaving a void in the plan drawing. While important for a holistic interpretation of the drawing, the structures associated with leaf edges in the graph are not used for semantic interpretation and we remove these edges before performing our semantic analysis of the graph structure. We also make a subsequent pass over the graph vertices to identify vertices that, after removing the leaf edges, have only two edges. These vertices are removed from the graph and the edges connected to these vertices are merged into a single graph edge. As an example, Figure 3 shows several leaf edges that have been pruned in Figure 4.

Semantic labels are assigned using a sequence of heuristics that assume the architectural complexes present in the image is a castle. We have a top-down approach for assigning face and edge labels that starts with large regions and iteratively decreases in scale until all of the edges and faces within the graph have been assigned a semantic label. This approach assumes that the drawing includes the entire castle complex and is based on observations of >150 castle plan drawings from a variety of sources (e.g., [Kau04]). The heuristics are stated as an ordered sequence of steps where, at each step, semantic labels for graph faces or edges are assigned. In our listing *italics* denote a semantic label associated with **bold** elements that come from the estimated graph $G(V, E, F)$:

1. The *castle surroundings* label (dark blue) is assigned to all **faces**, F_k , that include the boundary of the image.
2. The *castle outer walls* label (yellow) is assigned to all unclassified **edges**, E_j , that bound the faces found in (1) as part of the *castle surroundings*.
3. The *courtyard* label (blue) is assigned to the unclassified **face**, F_k , having largest area.
4. The *tower wall* label (orange) is assigned to unclassified **edges**, E_j , that are part of the *castle outer walls* class and extend significantly towards the exterior of the castle. Detecting towers is accomplished by comparing the curve fragment associated with the tower edge to a straight-line



(a)

Figure 5: The original archaeological plan drawing for the Crusader fortress at Apollonia-Arsuf in Israel. Results for this image are shown in Figure 6 (top row). The proposed method does not use information from some annotations such as the topographical lines in this plan drawing.

- curve fragment that connects the vertices spanned by the edge and applying Jensen's inequality; a test for convexity.
5. The *tower face* label (green) is assigned to unclassified **faces**, F_k , that include at least one *tower wall* as a bounding edge as specified in (4).
6. The *great hall* label (light blue) is assigned to the unclassified **face**, F_k , having largest area.
7. The *great hall wall* label (light red) is assigned to unclassified **edges**, E_j , that bound the *great hall* specified in (6).
8. If more than three faces remain unclassified, we assume the castle contains a chapel. The *chapel* label (cyan) is assigned to the **face**, F_k , having largest area from the list of unclassified faces.
9. If a chapel was found, the *chapel wall* label (red) is assigned to the unclassified **edges**, E_j , that bound the *chapel* specified in (8).
10. Remaining faces are classified as unknown (light green) and likewise for edges (white).

We also note that, in some cases there exist long thin white in plan drawings often associated with passageways and typically occurring in the gate area. As a pre-processing step, these faces are eliminated from the list of faces and marked as unknown regions (white).

4.7. Shortcomings of our Approach

While our initial system performs well for a number of Medieval castles, there are a number of potential sources of error which may occur:

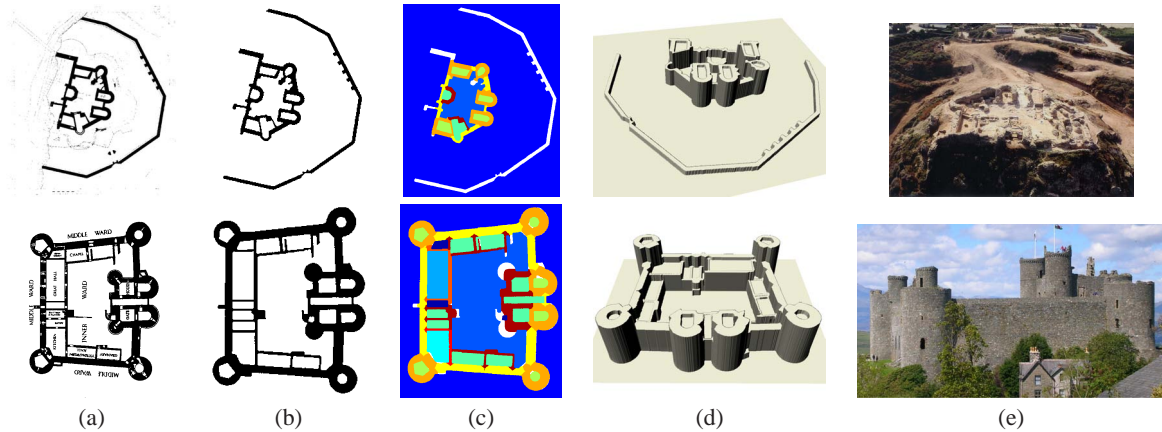


Figure 6: Results for two castles are provided as separate rows: (top) A Crusader fortress at Apollonia-Arsuf in Israel (bottom) Harlech castle in Northern Wales (see Figure 1) for a similar set of results computed for Caerlaverock castle in Southern Scotland. Colors shown in column (c) indicate the semantic parsing of the architecture (see § 4.6 1-10).

- Potential problems with the input image data,
- Errors in classification, i.e., limits to the generality of our heuristic rules,
- Errors in geometric accuracy

Typically a user will need to dedicate a short period of time (<5mins) to “clean up” the input image for our system (see § 4.1). Typical problems due to the input image data occur when:

1. the size of the castle within the image is either too small (< 10% of the image) or too big (extends off the image boundary).
2. the image includes other structures apart from the castle, e.g., the image contains a building separated the castle.
3. the castle is surrounded by another structure / fortification in the image, e.g., if present, the outer-bailey/enceinte of the castle must be manually removed.
4. the castle boundary is not a closed, thick contour in the image, i.e., the boundary is not distinguishable from other annotations in the image such as topographical lines, grid-lines, etc (see § 4.2 for details).

Note that issue (1) is a resolution-related problem and issues (2,3,4) are issues that relate to the topological structure of the graph extracted from the image (see § 4.4).

While our proposed method is completely new and works well for a number of castles it may incorrectly identify some parts of the castle. The generality of the heuristics applied in § 4.6 is also limited and applies generically *only* to fortresses and castles constructed within the Medieval period. While castles from other periods generally share the same structure, heuristics for internal structural complexes as specified in rules 6-10 are more likely to generate incorrect classifications for castles built before or after the Medieval period.

At the moment, no efforts are placed to extract topological, i.e., height information with respect to the “ground

plane,” which are present in some archaeological drawings (see Figure 5). Hence geometric errors may exist in the model due to two sources: (1) the exact topography of the ground in the vicinity of the castle is assumed flat and (2) the height of the castle structures is not available from the plan drawings, our approach assumes pre-defined heights and proportions for classified structures based in their class-label.

4.8. Generating the 3D Model

At the moment we generate the 3D model using a volumetric extrusion of the semantic labels estimated in the previous step. Semantically distinct regions are extruded to a different z-value with towers having the highest extruded offset, tower rooms having the second highest offset, followed by outer walls, great hall walls, great hall room, chapel walls, chapel room, unknown walls, unknown rooms, the courtyard and finally the castle surroundings which are placed in the $z = 0$ plane. More sophisticated models can exploit the semantic labels to generate detailed geometry using methods such as [MWH*06] to fill-in missing information procedurally in a manner appropriate to the identified semantic type.

5. Results

The parsing and recognition results are shown in Figures 1 and 6. In general, we are pleased with these results which reliably classify the castle surroundings face, the castle outer walls, the castle towers, tower rooms, and the courtyards for each of the shown examples. As demonstrated by comparing the actual buildings to those reconstructed automatically, the semantic labels provide important information that provide a good coarse estimate of the structure. Note that, at present, existing photographic information is not leveraged to augment the 3D model which is an area of interest for future investigation.

6. Conclusion

We have presented software that processes archaeological plan drawings for the purpose of identifying significant semantic sub-structures present within the drawing. While other methods exist for processing architectural documents, our approach differs from previous approaches in both goal and methodology. Our goal is to be able to process documents, old and new, that describe historic sites. Our approach generates semantic labels from scanned manuscripts using a process that is fully-automatic in most tested cases and in some cases requires a modest amount of user interaction. The shape-and-topology model afforded by using a skeleton model for shape and a graph model for topology allows heuristics to be defined that can identify important semantic structures within the plan drawing. Effective heuristics for medieval castles and fortresses were discussed that led to satisfactory classification of semantic labels for architecture of these structures and the rooms within these structures. The process results in several useful products including the final parse of the plan drawing into semantic labels. These products include:

- Software that provides fully-automatic methods for extracting architectural structures from digitized documents. If necessary, this process can be controlled interactively to efficiently extract structural data from plan drawings.
- A combined shape-and-topology model for building complexes. The shape component of the model represents the complex as a connected group of curve-elements where each curve element corresponds to areas where the walls are linear or cylindrical in shape which are shapes typical to medieval castle construction. The topological component of the model uses a simple planar graph having edges associated with each wall, vertices at wall junctions and faces for open spaces that bound the complex (including the castle exterior).
- Heuristics are used to assign semantic labels to each architectural curve-element and each open space within the image.
- A virtual 3D model of the architecture is estimated using the estimated semantic labels and the estimated binary image.
- An explicit list is provided that details issues that may cause erroneous outputs and issues that relate to the accuracy of the generated 3D model.

Since the 3D model is derived from plan drawings where the architectural contour is well-defined, the approach automatically provides near-pixel level accuracy at all locations which is very difficult and time-consuming to guarantee when manually constructing 3D models from the same drawing. Hence, these automatically-produced models can provide unprecedented accuracy to the *in-situ* remains not feasible with conventional manual model-building techniques.

While this article represents initial work on this topic with limited scope (castles/fortresses), we envision that subsequent enhancements to this method will be a valuable tool

for efficiently generating accurate 3D models for many different historic structures. This work represents a novel fusion of digital document analysis, semantic parsing of images and 3D reconstruction techniques as they apply in a cultural heritage context. There are several avenues for enhancement of the approach that include, interpreting other annotations available in typical plan drawings, using a procedural model to automatically generate local building details based on the generated semantic labels, integrating information from vertical cross-section drawings and extracting information from other archaeological annotations that are currently removed as image noise. We are also interested in examining how the heuristic aspect of this approach can be generalized to include different structures or possibly exchanged for a more sophisticated approach that involves pattern recognition.

References

- [DL99] DORI D., LIU W.: Sparse pixel vectorization: An algorithm and its performance evaluation. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 21, 3 (1999), 202–215.
- [DTASM00] DOSCH P., TOMBRE K., AH-SOON C., MASINI G.: A complete system for the analysis of architectural drawings. *International Journal on Document Analysis and Recognition* 3 (2000), 102–116.
- [HZ04] HARTLEY R. I., ZISSERMAN A.: *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press, 2004.
- [Kau04] *The Medieval Fortress*. Da Capo Press, 2004.
- [Ley08] *Applications of Medial Symmetry Representations of Shape*, vol. 37. Springer Netherlands, 2008, ch. From the Infinitely Large to the Infinitely Small, pp. 327–351.
- [LMV01] LIADOS J., MARTI E., VILLANUEVA J. J.: Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 23, 10 (2001), 1137–1143.
- [LS92] LAM L. S.-W. L., SUEN C. Y.: Thinning methodologies—a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 9 (1992), 869–885.
- [MVW*06] MULLER P., VEREENOGHE T., WONKA P., PAAP I., GOOL L. V.: 3d reconstruction of puuc buildings in xkipche. In *The 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage* (2006), pp. 139–146.
- [MWH*06] MULLER P., WONKA P., HAEGLER S., ULMER A., GOOL L. V.: Procedural modeling of buildings. In *Proceedings of ACM SIGGRAPH* (2006), pp. 614–623.
- [Soi03] SOILLE P.: *Morphological Image Analysis: Principles and Applications*, 2nd ed. Springer-Verlag, 2003.
- [Tau91] TAUBIN G.: Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 11 (1991), 1115–1138.
- [TCYZ03] TU Z., CHEN X., YUILLE A. L., ZHU S. C.: Image parsing: Unifying segmentation, detection, and recognition. In *Ninth IEEE International Conference on Computer Vision* (2003), pp. 18–25.
- [YWR09] YIN X., WONKA P., RAZDAN A.: Generating 3d building models from architectural drawings: A survey. *Computer Graphics and Applications* 29, 1 (2009), 20–30.