INAUGURAL-DISSERTATION
zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Dipl. -Arch. Pheakdey Nguonphan
aus Prey Veng, Kambodscha

Tag der mündlichen Prüfung: 09. Januar 2009

Computer Modeling, Analysis and Visualization of Angkor Wat Style Temples in Cambodia

Gutachter: Prof. Dr. Dr. h.c. mult. Willi Jäger
Prof. Dr. Dr. h.c. Hans Georg Bock

Zusammenfassung

Das archäologische Gebiet von Angkor und die gigantischen Steintempel sind die verbliebenen Hauptquellen zu den fast vergessenen alten Khmer-Zivilistation die einst ihre Blühtezeit zwischen dem 7. und 13. Jahrhundert hatte. Hunderte von Tempeln wurden erbaut, doch mit der Zeit sind sie fast in Vergessenheit geraten, und haben erleidete sehr unter dem tropischen Klima, Kriegen aber auch Plünderungen gelitten. Die meisten von Ihnen liegen nun völlig in Trümmern, und Viele sind teilweise zerstört oder im Einsturzgefahr. Inzwischen wurde Angkor als UNESCO Weltkulturerbe erklärt und ist eines der beliebtesten Tourismus-Ziele, aber auch für Wissenschaftler attraktive die neue Methoden zur Unterstützung der Restaurierung, Bewahrung und Tempel-Rekonstruktion erforschen.

Die Buddhistischen und Hinduistischen Tempel von Angkor wurden auf Basis der heiligen Lotusblüte, die in beiden Religionen den Mikrokosmos darstellt, als einem der Grundsymbole erbaut. Sechs Hauptdarstellungsformen des Lotus wurden identifiziert, die in einer bestimmten Weise kombiniert, und fast auf allen Tempelelementen abgebildet sind. 3D Computer-Rekonstruktionen von solchen Tempelelementen sind schwierig und enorm aufwändig.

Diese Dissertation befasst sich mit der Analyse architektonischer Funktion und Geometrie der Lotusmotive und dem Versuch die Kombinationsregel der sechs Lotusmotive zu entschlüsseln. Um dies zum Einsatz zu bringen werden, mit Hilfe mathematischer Modelle und bestimmter Algorithmen eine neue modul-basierte Rekonstruktionsmethode und zwei weitere Methoden als Ergänzung speziell für Angkor Temple entwickelt.

Das Ergebnis stellt ein in dieser Dissertation entwickeltes Computerprogramm dar, das drei verschiedene Rekonstruktionswerkzeuge umfasst und mit dem hoch detaillierte 3D Modelle einfach und schnell erzeugt werden können. Schließlich werden mehrere 3D Modelle von ausgewählten Tempeln als Endergebnis vorgestellt.

Schlagworte: Module-based 3D Computer Reconstruction, Image Processing, Virtual Reality, World Heritage in Digital Age, Virtual Museum, Architectural concept of Angkor Temples.

Summary

Angkor archeological site and its gigantic stone temples are the major links to a nearly lost civilization of the ancient Cambodia that once flourished during the 7th and 13th century. Hundreds of temples have been built, however, over the years, the vegetations attacks, wars, and lootings caused many temples to totally decay, some are partly destroyed or in danger of collapse. Just as Angkor archeological site and temples were finally registered as the *UNESCO World Heritage Site in Danger* in December 1992, local teams and international scientists with professional expertise from various fields began doing their researches seeking for evidence of the hidden ancient Khmer culture. New efficient methodologies for preservation, restoration and reconstruction of the temples of Angkor are of the great demands.

Angkor temples are religious buildings dedicated to Buddhism and Hinduism. The sacred lotus which itself symbolized the microcosm in both religions is an essential element of the architectural concept of Angkor temples. Six forms of lotus flower motifs are depicted as ornament based on a particular structuring rule, and are found almost on the entire temple elements. 3D computer reconstruction of these elements using conventional methods available in CAD tools is a very difficult and time consuming task.

This research introduces new module-based computer reconstruction methods which makes use of the structuring rule of the lotus motifs. Mathematical models are defined to describe six modules and their properties. Explicit definition of the module structuring rule and the conditions required for its implementation are discussed in details. Finally three different sets of algorithms for three distinct reconstruction approaches are defined.

As a result, a new software package is developed which covers three different reconstruction tools. This computer program is user-friendly, and provides highly detailed 3D models of Angkor temples with a few mouse clicks.

Keywords: Module-based 3D Computer Reconstruction, Image Processing, Virtual Reality, World Heritage in Digital Age, Virtual Museum, Architectural concept of Angkor Temples.

Acknowledgments

This is a great opportunity to express my respect and thanks to my supervisors, Prof. Dr. Dr. h.c. mult. Willi Jäger, and Prof. Dr. Dr. h.c. Hans Georg Bock, for giving me the chance to take part in this scientific facility, and for providing me precious ideas and advices to make this Ph.D. research possible. My special thanks are devoted to Prof. W. Jäger who helped me so much with mathematical modeling, and for putting his full trust in me and my work. This research was financially supported partly by *Gottlieb Daimler- and Karl Benz Foundation* and the work group of Prof. H. G. Bock, and the main support was from the work group of Prof. W. Jäger. I would like to express my deepest thank to all sponsors.

The *Angkor Project Group* program and activity also help strengthen my knowledge in understanding the Angkor temple architecture as well as in 3D modeling experiences. Thus again I want to thank Prof. H. G. Bock for initiating this project as well as many other scientific activities relating to world heritage and Angkor. At this point I have to say thank you to Dr. Michael J. Winckler for giving me a lot of chances to practice myself for this research, and to train students in their software practical courses. He also supported me with programming advices and discussions on mathematical issues. I am especially pleased to thank Dr. Susanne Krömker for always keeping her office door opened for any open question. She was very much involved in getting this research successful.

I want to say thank you to my colleagues and friends in the *Applied Analysis, Simulation and Optimization* as well as the *Computer Graphics* group for their friendship cooperation. Special thanks go to Dr. Johannes Schlöder for his thorough proofreading and very fruitful discussions. I thank to my friends Somporn Chuai Aree and Dr. Elfriede Friedmann for always being ready for constructive talks. I am so grateful to Mr. Sareth Lek, *Faculty of Architecture in Phnom Penh, Cambodia*, who sadly passed away in 2003. He initiated this project together with Prof. H. G. Bock.

This dissertation is dedicated to my mother Som Din, my father Prof. Nguon Phan, my brothers and sister, and thank for their endless love, encouragement and supports in any aspect. This work is surly devoted to my wife and my lovely son who have given me so much power to keep moving forward.

Contents

Acronyms

ANU	Australian National University
APSARA	Authority for the Protection and Management of Angkor and the Region of Siem Reap
ATE(s)	Angkor Temple Element(s)
ATD	Angkor Temple Detector
ATG	Angkor Temple Generator
AWST	Angkor Wat Style Temple
CAD	Computer Aided Design
CSG	Constructive Solid Geometry
EFEO	Ecole Française d'Extrême Orient
IDE	Integrated Development Environment
LPE	Lower Part Element
SSP	Scale Symmetric Property
UNESCO	United Nations Educational, Scientists and Cultural Organization
UPE	Upper Part Element
VBA	Visual Basic for Application

List of Selected Symbols

Symbol	Meaning	Defined in
a	first module (module vector)	section 2.2.2
b	second module (module vector)	section 2.2.2
c	third module (module vector)	section 2.2.2
d	fourth module (module vector)	section 2.2.2
$d(v_t)$	degree of vertex v_t, $d(v_t)=d_{in}(v_t)+d_{out}(v_t)$	section 2.3.2
$d_{in}(v_t)$	in-degree represent arrows pointing into v_t	section 2.3.2
$d_{out}(v_t)$	out-degree represent arrows pointing out from v_t	section 2.3.2
e	fifth module (module vector)	section 2.2.2
e^t	end point of t	section 2.3.1
e^{p_t}	end point of module profile p_t	section 2.3.1
E	set of edges of G	section 2.3.2
E'	set of edges of W	section 2.3.2
f	sixth module (module vector)	section 2.2.2
$f_i(x,y)$	function to describe p_t	section 2.3.1
$G(V,E)$	graph of principle rule for reconstruction of S_{ATE}	section 2.3.2
$\lvert G\rvert$	order or number of vertices in G	section 2.3.2
$\lVert G\rVert$	number of edges in G	section 2.3.2
h	height of the ideal element profile P_{ATE}	section 2.3.2
h_0	a selected point that forms the $edge(s^t,h_0)$	section 2.3.1

M	central point of arc segment in p_t	section 2.3.1
N	number of t in S_{ATE} and length of walk W_N	section 3.1.1
n_t	number of type t (coefficient of term $n_t t$)	section 3.1.1
P_{ATE}	2D profile of Angkor Temple Element (ATE)	section 2.3.2
p_t	2D profile of module type t	section 2.3.1
S_{ATE}	sequence of modules in an ATE type	section 2.3.1
s^t	start point of t	section 2.3.1
t	module of a certain type $t \in \{a,b,c,d,e,f\}$	section 2.3.1
V	set of vertices of G	section 2.3.2
V'	set of vertices of W	section 2.3.2
$W_N(G)$	walk of G with length N	section 2.3.1
x_e	x coordinate of the end point e^t	section 2.3.1
x_s	x coordinate of the start point s^t	section 2.3.1
v_t	vertex of G from type t	section 2.3.2
y_e	y coordinate of the end point e^t	section 2.3.1
y_s	y coordinate of the start point s^t	section 2.3.1
$\alpha_t(s^t, h_0, e^t)$	angle value of module vector t with $edge(s_t, h_0)$	section 2.3.1
$\varepsilon = [v_i, v_{i+1}]$	directed edge of G	section 2.3.2
$\sigma_{i,t}$	vertices of module profile p_t with $i = 1,...,n$	section 2.3.1
σ_1, σ_N	start point and end point of p_t respectively	section 2.3.1
ς^t	scale factor of t	section 2.3.2
$\lambda(\sigma)$	line segment or edge of module profile p	section 2.3.2
$\mu(\sigma)$	arc segment of module profile p	section 2.3.2

Chapter 1

Introduction

Historic architectures, especially religious buildings are constructed based on specific theoretical and practical principles. Angkor temples, as state temples of the *Khmer* (Cambodian) empire are rich of conception and iconographic representations. A detailed description about this will be presented in Chapter 2. The first section of this chapter presents an overview on the history of Cambodia relative to the Khmer temples and the origin of its architecture. In the second part we bring out the essence of this research including problem statement, review of related works and motivation. Finally we present the contents of each chapter in this dissertation.

1.1 Angkor Temple Origin and Architecture

This section describes a brief history of Angkor temples and the advantages of their architecture for this research. In the first part we will describe the establishment of the state of Angkor and point out the reason why it has become the world's most concentrated area with historic temple complex as well as its attractions to scientists and tourists at the present time. In the second part we present properties of Angkor temple architecture which is the core of our discovery, and that plays the central roll for the solution.

1.1.1 History of Angkor Temples and Cambodia

In around the first century of the Christian era, Indian traders extended their international business towards Southeast Asia, seeking for new exotic products. They took advantage of the tropical monsoon wind that pushed the Indian merchant fleets back and forth from India to the southern part of the first known Khmer empire, called *Funan,* locating in today's southern Vietnam.

Later the Brahmans, priests and scientists of the highest social class from India came with the traders, and brought their principle religion, Hinduism and Buddhism [Sti79] to Cambodia. The Khmers warmly welcome the Indian people as well as their culture, and skillfully converted the religious doctrines into significant architectural concepts. As soon as the Khmers has developed its own language based on the Indian *Sanskrit* in about the 6th century A.D., the first registered Khmer temple arose, the *Phnom Da* temple (figure 1.1.1), situated in *Angkor Borei,* the southern of the modern Cambodia, in *Takeo* province.

After the royal families of Funan from *Lower Chenla* and *Upper Chenla* became unified at the end of 8th century, the Khmer empire was established.

Rice was one of the most important local products, and played a significant roll in development of the empire. The Khmers gradually found out that rice could be best grown around the area of the *Tonle Sap,* a great lake with unique natural properties that provides optimal conditions for agricultural development.

In 790, the young prince *Jayavarman II* revolutionary created the *devaraja* cult (Khmer language: *Tevareach*), which literally means "king of the gods", and published it for the first time in 802 on stone. He established a capital closer to the great lake, near the *Mahendraparvata* mountain (today's *Kulen* mountain, about 30 km to the east of the present Siem Reap province), and later in *Roluos*. His devaraja cult and great achievement in territorial expansion immensely inspired his successors, and might have been the key factors that brought glory to the empire. Although his capital of Rolous does not correspond to the well-known *Angkor area* we are referring today, Jayavarman II had opened a new era of the Khmer history, and is generally accepted to be the founder of Angkor and Angkor period [Ort06].

Figure 1.1.1
Phnom Da, The first Khmer temple mountain built around 550 A.D., constructed in the former Khmer capital *Funan.*

Angkor is a Khmer word, derived from Sanskrit *nagara* and means "city". The Angkor period lasted from 802 to 1244 and covered a successive reign of 20 different Khmer kings ([Roo01] 39 kings) in which the history memorized the summit of Khmer culture and glorious achievements.

In the sense that a king is the king of gods, he is obliged to make his capital into the home of god. Thus the Khmer kings constructed gigantic temples based on the Indian divine design and other religious iconography as a microcosmic representation and the central axis of the universe. This conception was best expressed in the world largest stone temple of *Angkor Wat*, which means *the temple of the city*, built during 1113-1150 by King *Suryavarman II* (see figure 1.1.2). The most outstanding king, not only in military but particularly for temple construction, is King *Jayavarman VII* (1181-1220). During his 39 years reign, he almost constructed as much temples as his predecessor all together including his state temple of *Bayon*.

A total number of 47 major temples have been built during Angkor period, and very few of them outside the Angkor area probably because of the advantages of the great lake. Figure 1.1.3 shows the Khmer temple increment from the 6th to the 15th century. From the time Jayavarman II unified Khmer kingdoms, temple construction continually grew, and reach the highest gradient in the 13th century during the reign of Jayavarman VII until the beginning of the 14th century. For detailed information, see the chronology in page 181.

Figure 1.1.2
Angkor Wat, 1140 built by Suryavarman II. in Angkor region.

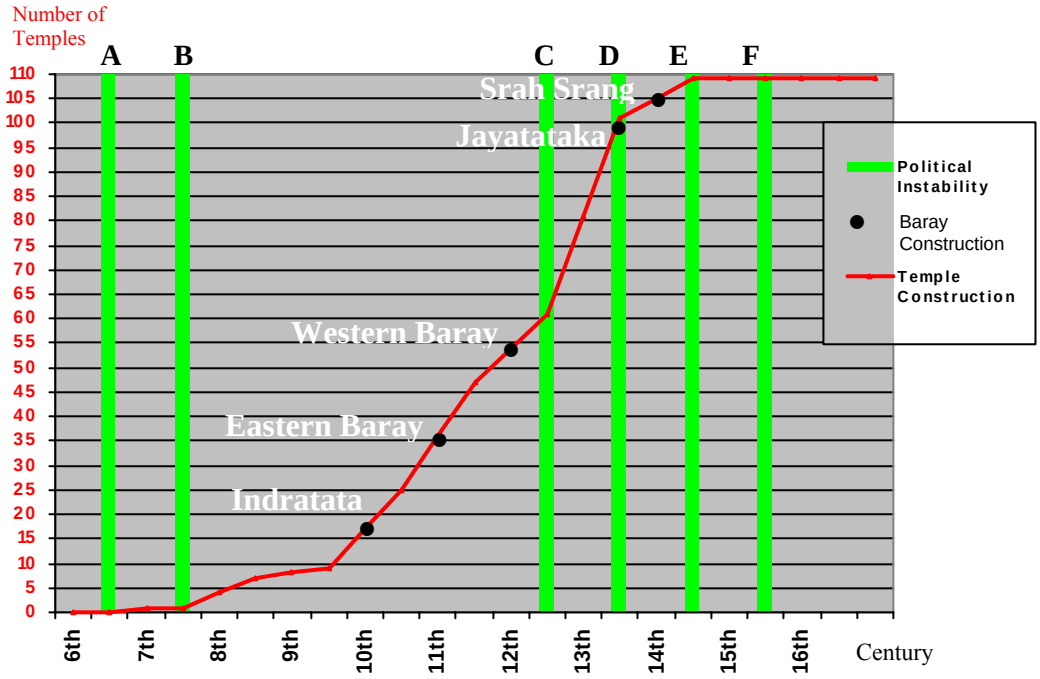


Figure 1.1.3.
Growth function of Khmer temple (major temples) construction during 6^{th} – 14^{th} century.

A Decline of Funan
B Split of Chenla
C Chamese attack Angkor
C < Golden age of Khmer empire < **D**
D Religious conflict and struggle within royal families
E Invasion by Ayutthaya
F The fall of Angkor, Siamese dominates Angkor

The frequent invasive attacks from the neighboring countries, the Siam and Champa (present Thailand and Vietnam respectively) finally led to the end of the

glory and the stone temple construction. In 1431, the Siamese troop of *Ayutthaya* defeated the decisive victory, and dominated Angkor for about 100 years [Lec14] while the Khmer King *Ponhea Yat* was forced to move his capital to Phnom Penh, where it is until today.

Latest Khmer temple records show that more than one thousand Khmer temples had been built during these ages, but over the years of wars but also looting and attacks from tropical climate and vegetation most of the temples are nowadays in very critical circumstances. Many of them are literally totally in ruin whereas some are in danger of collapse and erosion. In 1991, the *UNESCO* established their regional office in Cambodia in order to assist the local authority *APSARA* to call for international help on safeguarding Angkor. Angkor was registered in the UNESCO World Heritage list as *World Heritage site in Danger* in December 1992. Since then, high-level national and international authorities open access for professional teams worldwide to conduct a wide variety of scientific and cultural research projects, with ambitious target for restoration, reconstruction, conservation, and developing the Angkor archeological park and its ruins.

With our experiences in traditional Khmer arts and architecture, we want to contribute in the field of reconstruction, restoration and documentation of Angkor temples. The challenge is to find out the advantage of Khmer temple architecture and combine it with available associated techniques in applied mathematics and computer science in order to support the 3D computer reconstruction of Angkor temples. In the next section we present one of the major architectural principles of Angkor temple.

1.1.2 Sacred Lotus as Architectural Concept

We have mentioned that Angkor temples are religious building, dedicated to Buddhism or Hinduism. In both religions, one of the most outstanding symbolisms in iconography is the lotus flower.

Since the ancient time of Hindu tradition, lotus blossoms are regarded as divine symbols for the creation mythology. The Hindu *god of creation* Brahma is often depicted as sitting on an opened lotus flower or with a lotus in his hand [Sch02]. Similar illustrations can be observed from Buddha. In Buddhism, lotus flower that is

floating above muddy water of desire and attachment represents purity of mind and body. Lotus symbolisms are also attached with the Buddhist and Hindu divine design of *Mandala* which is the fundamental design principle for temples constructions. We are not going to discuss in detail about the essence of lotus symbolism in religious iconography. For more information, we recommend to refer to the very well written publication of Fredrick W. Bunce [Bun02].

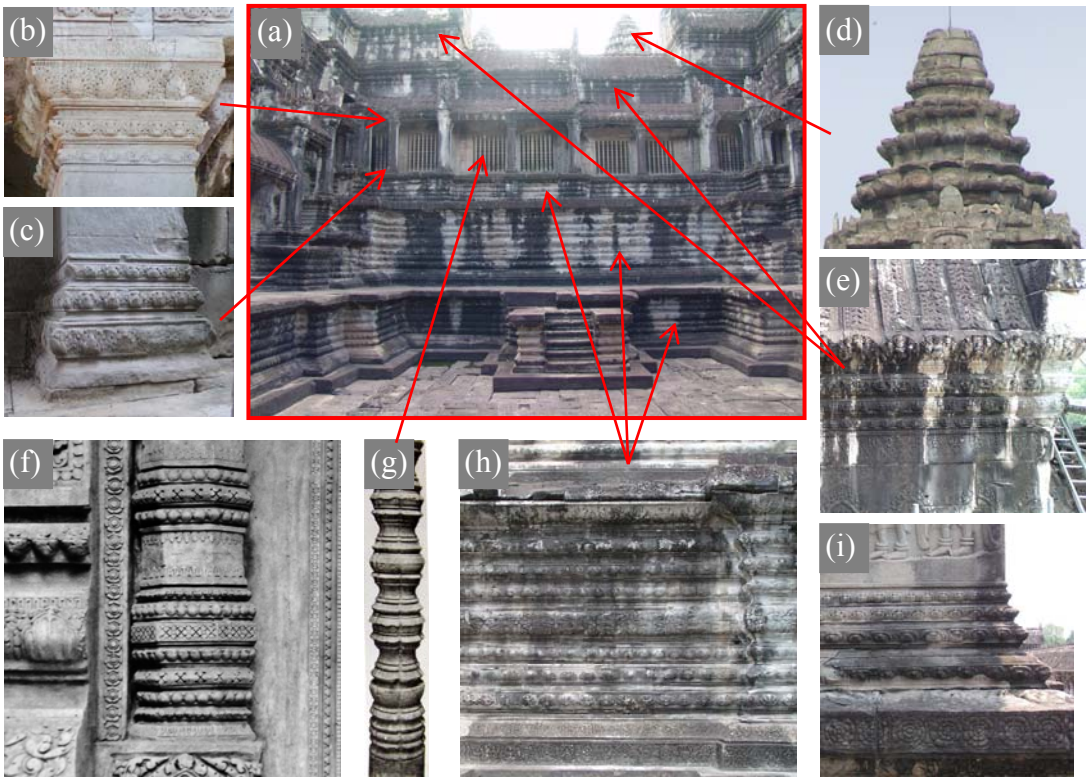


Figure 1.1.4
Architecture of Angkor temple elements shown on ((a), image by ANU) a part of Angkor Wat cruciform of the second level gallery, including (b) column-capital, (c) column-base, (d) crown of the tower, (e) wall-capital, (f) colonnette of an entrance, (g) window grid, (h) basement and (i) wall-base.

The ancient Khmer temple architects and artists must have been very much inspired by the theory and practices of the sacred lotus ideology. They skillfully converted this concept into a unique work of art and architectural element, and gave the Khmer temple structure an unmistakable characteristic. Beside constructing the temple towers in a form of a closed up lotus flower, the architects illustrated on almost any temple element with various physical properties and positions of the lotus

petals. Figure 1.1.4 shows some examples of this application. A detailed description is given in section 2.2.2.

These rich ornamental details characterize the essential aesthetic of Khmer art and temple architecture but elements with such details are extremely difficult to model using conventional CAD modeling tools. Commonly the details are simplified or ignored, and a primitive geometric form of the element is reconstructed instead. In order to obtain a realistic representation of the lotus motif, an available photo of the element is used as *texture* to map on the primitive geometric model.

Such simplified 3D models require less computing task for visualization, and thus facilitate the modeling process as well as faster to render however, as a consequence is we loose a lot of valuable architectural information of the temple. Besides, we obtain only a very limited quality of the virtual reality representation. In the next section we introduce the main problem in modeling Angkor temples and how we aim to solve it.

1.2 Scope of Research

In the first part of this section we present the literature reviews in which distinct techniques for 3D modeling and 3D reconstruction especially the reconstruction of Khmer temples are discussed. For each related work we will give some comments on the advantage and disadvantage of the technique for application. The second part of this section we describe a suggested method for reconstruction of highly detail 3D model of Angkor temples. We also show the problem formulation and our motivation for doing this dissertation. Finally, present the overall structure of this thesis.

1.2.1 Related Works and Motivation

The main subjects in the works that we are going to present now, and which are of our interest are the techniques for 3D computer reconstruction of Angkor temples and the effectiveness of their applications.

There are a huge number of researches concerning conservation and restoration of the historic park of Angkor and its temples. Many of them are trying to

get the best use of the state-of-the-art technologies and tools in order to better understand the origin of the ancient Khmer culture, art and architecture. The work of *Sonnemann, Sauerbier, Remondion* and *Schrotter* [SSRS06] for example uses the aerial images to reconstruct 3D models of Angkor temples. The information obtained from GIS provides a very precise measurement of the ground floor plans of the existing temples and other visible horizontal outlines of its elements. Although many temple elements could not be depicted because they are hidden from the top view and the 3D model could only be observed from a limited distance, this method is ideal for visualization of macro-scaled models such the temple complexes of Angkor area. We also observe that most of architectural information and details on the vertical surfaces of the temples are lost.

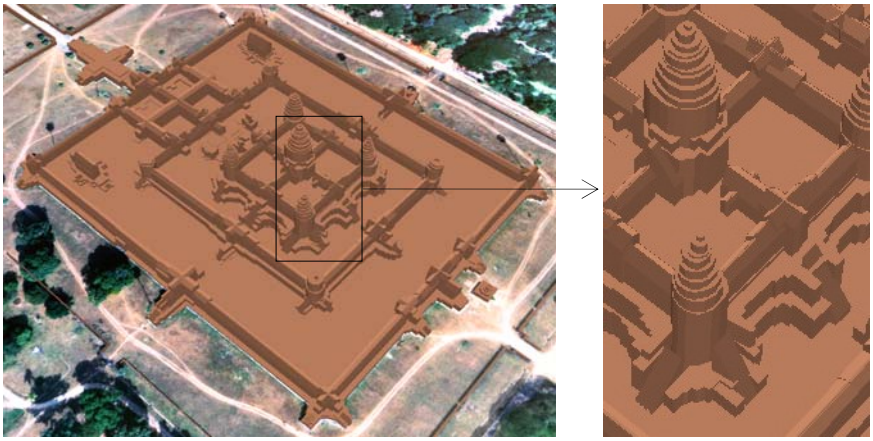


Figure 1.2.1
3D model of Angkor Wat temple by extruding the horizontal outlines of each element captured from satellite images and additional height measurements [SSRS06].

Similar 3D model quality was achieved in the post-graduated work of *Cunin* [Cun00] where an aim is, to visualize the 3D complex structure of the Bayon temple together with links to the database of the historical and architectural descriptions of the Bayon and its elements. A limited level of detail of the 3D geometry is presented as a result (figure 1.2.1). The author uses the manual modeling method that is available in the conventional CAD tools for the 3D reconstruction which would be too much time-consuming to model all the details of the lotus motifs. Nevertheless, the result is suitable to apply for the purpose of virtual museum.

More virtual reality oriented 3D visualization of Khmer temple can be seen in the award winning (SIGGRAPH2001) project of *Levy* [Lev01]. He focuses on reconstruction of the whole *Phimai* temple complex, locating in Thailand, with considerable high level of detail of the model geometry. Compared to the previous works, the result here is much higher detailed, and suitable to use for the 3D virtual realistic simulation. In image (a) of figure 1.2.2 we see the detailed 3D geometry presented with the texture of the corresponding element mapped on it.

We observe that still in this example the details of the geometry do not correspond properly to form of the lotus motifs. This partly causes the mismatch between the geometry and the texture.

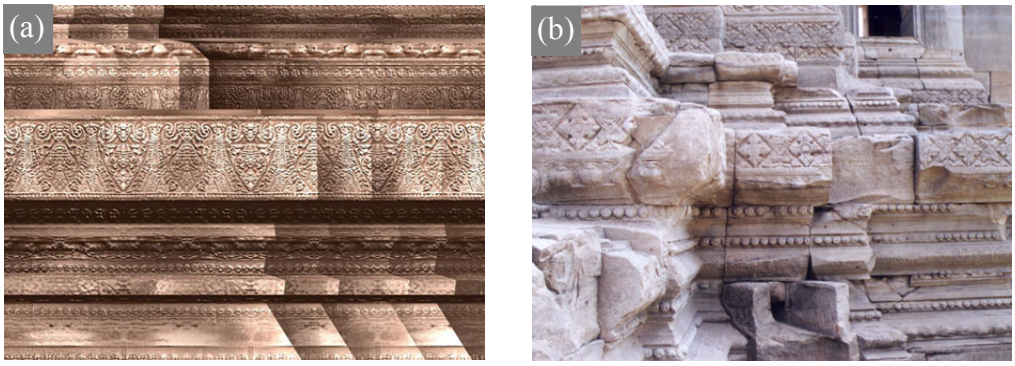


Figure 1.2.2
Details of Phimai temple basement: (a) computer reconstruction, (b) photography of the basement [Lev01].

The more detailed the 3D geometry of a model is, the more realistic it appears. The best method for 3D reconstruction, or rather, recording an existing object is using the 3D laser scanning technique. A Japanese team confirms the effectiveness of such application in their research on *Significance of Digital Reconstruction of Historic Buildings Using 3D Laser Scanner* [YTSN03] by scanning one of the largest objects ever scanned ─ the *Prasat Suor Prat N1* tower in Angkor.

The precise 3D scanner records every unevenness on the temple surface including its decay, cracks, lost parts but also the still existing architectural details. The disadvantage of this method is among others the huge dataset of the recorded 3D *point clouds* on which later the triangulation is based. The data increment and the difficulty in scanning inaccessible parts limit the scale and complexity of the object

for scanning. On the other hand, the resulting objects are meshes instead of solid objects which hardly allow us to examine or edit a specific part of the whole structure. However, for analyzing the physical condition and recording an existing object such as the Prasat Suor Prat tower, it is the best method.

Thus the choice of method and technique depends on the field of application. Our research is focusing on 3D solid modeling of Angkor temples that is based on CAD tools. Each temple element will be individually reconstructed as solid objects and is therefore easy to handle for modifying the geometry, analyzing the temple structure, and are virtual reality oriented.

We have seen from the literature review about the difficulties in modeling the temple elements with lotus motifs. Our motivation is to develop new approaches for highly detailed 3D reconstruction of Angkor temple elements with lotus motifs. We put our main concentration on preserving the original form of the lotus motifs and accelerating the reconstruction and modeling process. We are also keen to provide possibilities to reconstruct temple elements in there ideal state but also based on available input data such as photography of the elements.

Before we present the scope of this dissertation, we firstly need to introduce our suggested method for 3D reconstruction of the Angkor temples using CAD tools.

1.2.2 3D Computer Reconstruction of Angkor Temples

The procedure for the computer 3D modeling of Angkor temples is suggested to follow the construction procedure at the construction site. We begin with tracing the form of the ground floor plan, see figure 1.2.3 image (a) for instance, which embeds information of the elements including the basements, columns, walls, windows, stairs, colonnette, etc. where the horizontal projection, the thickness and number of each element is known. From this 2D information we can generate a 3D model by extruding the outline of each element by its corresponding height which can be read from the elevation.

In figure 1.2.4 we demonstrate this method on 3D modeling of the main basement of the southern library building at the second level of Angkor Wat temple

complex without lotus motif. Using this method we obtain a 3D model of the same quality as in [SSRS06] since it is the same procedure except that for our example we use a 2D drawing of the ground floor plan instead of satellite images.

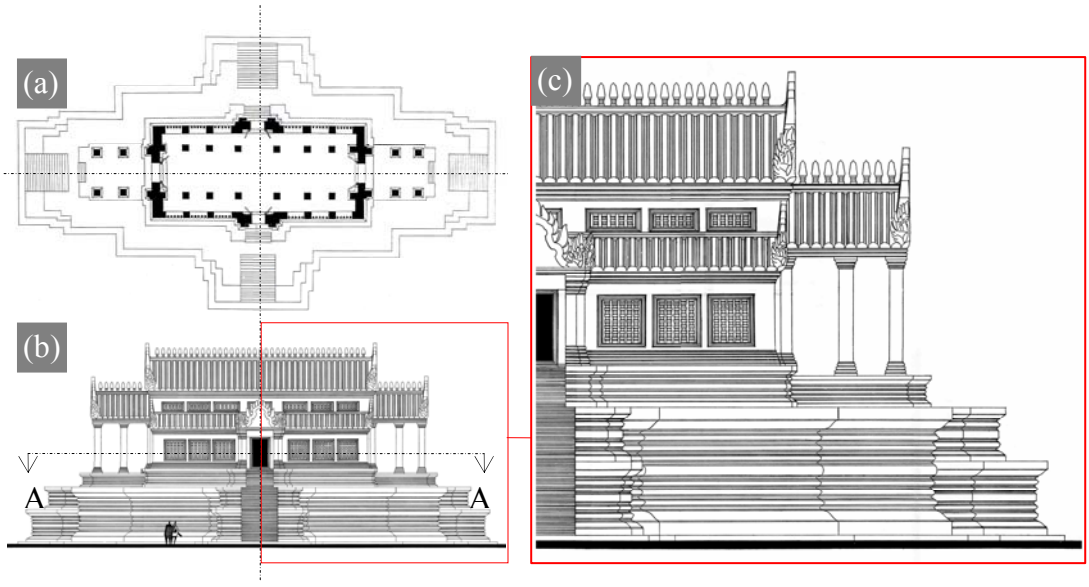


Figure 1.2.3
Plans of the southern library on the 2nd level of Angkor Wat complex: (a) ground floor plan; obtained by a horizontal cross-section (AA) shown in (b), (b) south elevation and (c) enlargement of the western part of (b). Drawings done by EFEO.

Highly detailed 3D computer reconstruction of Angkor temples with lotus motif molding require the 2D information of the lotus motif profile. To create the 3D models we need to *extrude* the 2D lotus motif profile of each element a long its traced ground floor information. For our reconstruction sample of the library we have sufficient information of the required details which can be gained from the image (a) of figure 1.2.3. The black bolded line profile in this cross-section plan of a basement gives us the relevant information of the lotus motifs of that element. Our suggested procedure for the 3D reconstruction of the basement with the lotus motifs is shown from image (b) to (f).

We firstly have to vectorize the profile of the element, presented as the red colored line profile in image (b). The obtained profile must be converted into a *closed line* sequence or a *region* (defined in AutoCAD system), see image (c). This region can now be extruded along a given *extruding path* which represents the ground floor projection of the basement.

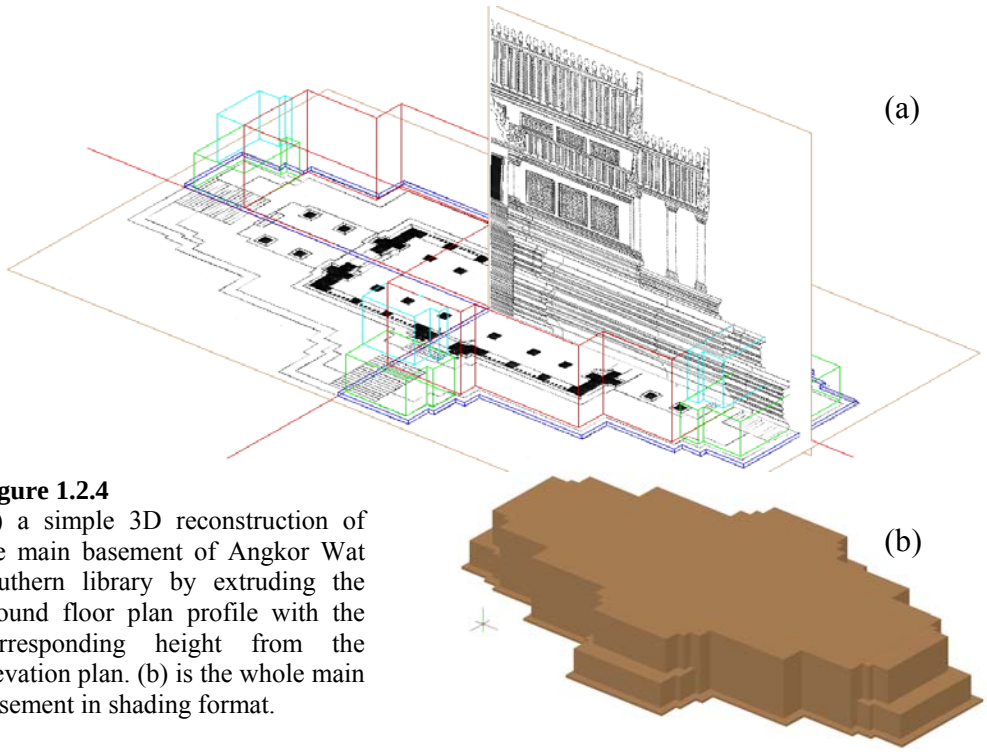


Figure 1.2.4
(a) a simple 3D reconstruction of the main basement of Angkor Wat southern library by extruding the ground floor plan profile with the corresponding height from the elevation plan. (b) is the whole main basement in shading format.

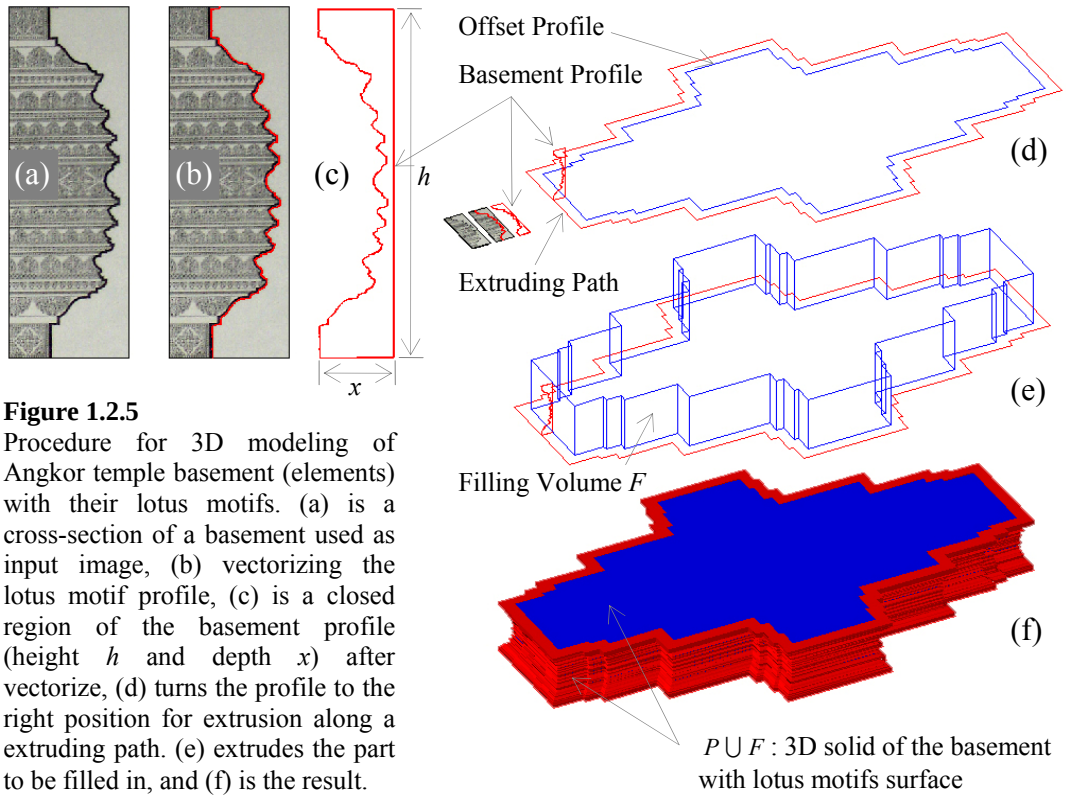


Figure 1.2.5
Procedure for 3D modeling of Angkor temple basement (elements) with their lotus motifs. (a) is a cross-section of a basement used as input image, (b) vectorizing the lotus motif profile, (c) is a closed region of the basement profile (height h and depth x) after vectorize, (d) turns the profile to the right position for extrusion along a extruding path. (e) extrudes the part to be filled in, and (f) is the result.

We obtain a pipeline alike 3D solid P of the profile in (c) alone the outer vertical surface of the basement. Finally we need to fill in the inner volume F simply by a vertical extrusion of the *offset* profile (offset by a depth of x) of the original extrude path, see (d), (e) and (f). A union of $P \bigcup F$ provides a complete detailed 3D model of the basement. The result of implementing this method can be seen from figure 1.2.6 (a) and (b). The detailed 3D geometry must be finally mapped by the corresponding texture of the element in order to illustrate the very fine details of the lotus motif as shown in the image (c).

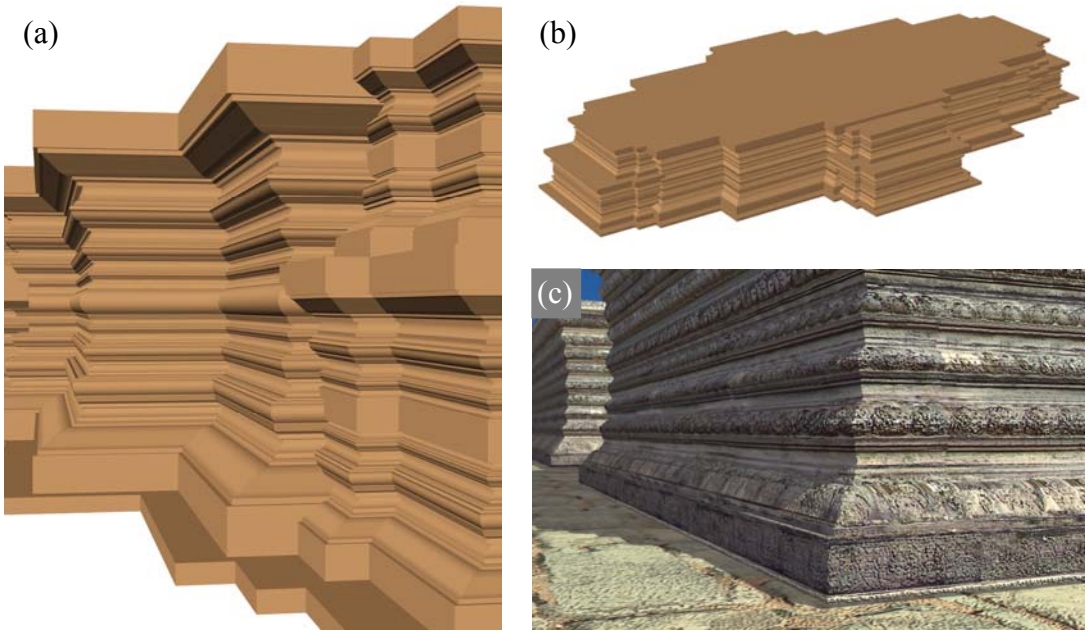


Figure 1.2.6
3D reconstruction of the basement of the Angkor Wat temple library shown in figure 1.2.1. (a) is an enlargement of the complete 3D geometry shown in (b). (c) is the final CG representation of the basement with the corresponding texture.

The most important and difficult phase in the procedure we have presented above is the reconstruction of the element profile with lotus motifs in step (b). The final quality of the element 3D model mainly depends on the quality of the reconstructed 2D profile. The problem is that every temple element with this characteristic is unique in the way of composing of the lotus motifs and proportion. Moreover, the process of vectorizing these profiles must be done manually which is a very time consuming process.

1.2.3 Contribution to the Reconstruction Methodology

Our overall field work is a sort of *Computational Architecture* which is a combined field of historic architecture, computer science and applied mathematics. We study the characteristics of the ancient Khmer temple arts and architecture and try to find out a certain architectural pattern in their structure. The mathematical modeling of the pattern makes possible to simulate and reconstruct the temple architecture using the *computer aided design* and computer graphics.

For the reason that there are several styles of Khmer temples, we will confine our study on Angkor Wat style temples which have been constructed during the reign of the King *Suryavarman II* (1113-1150 AD).

The major part of this dissertation covers the discussion on a *module-based computer reconstruction* of Angkor temples. We will begin with a successive refinement of the complex temple structure down to the individual components (lotus motifs) of the temple elements. Then we identify the types of the lotus motifs and categorizing them into a finite set of types of the components, and try to unlock the hidden rules for structuring them. Based on these rules, in the inverse direction, we will be able to reconstruct the complex structure by reassembling stepwise the components back to an element, and then further assemble the elements to get a temple complex.

The structuring rule is mathematically represented by a specific adjacency matrix which is responsible for generating a proper sequence of the lotus motifs in an element structure. Proportional measurements of the components from the photography of a set of selected temple element will be carried out. From the gained data we will setup predefined ideal mathematical functions for the management of the proportions of the components in the sequence. With the associated algorithms a new automated reconstruction method will be presented that is specified for the reconstruction of highly detailed two-dimensional profiles of the elements of Angkor and Khmer temples quick and easy.

The inverse problem is presented afterwards. By using an image analysis technique and computer graphics we will be able to generate an additional reconstruction method that could automatically detect the element profile from an

input image of the temple element. Because this method is apart from predefined modules and construction rules, it can be implemented to reconstruction not only Angkor and Khmer temples but also to reconstruct any building of which the element surfaces have the properties of moldings.

As the final result, we develop a new software tool which is specialized for high quality 3D modeling and reconstruction of Angkor temples. The development of this program is aimed to provide possibilities not only for reconstruction elements of an existing temple, but can also be used for experimental purposes for developing new Khmer temple design concepts based on ancient Khmer construction rule. The results are also supposed to contribute in seeking for the identity of Khmer culture and temple architecture in order to support cultural heritage of Angkor, which has always been standing as a symbol of Cambodia since about a thousand years ago.

1.2.4 Outline of Dissertation

This dissertation is divided into six chapters. We firstly provide an overview on the history of Angkor and Angkor temples, and we will point out the overall problems that inspired our research. A literature review of the related works including problem formulation, a motivation for this thesis, and the methods to solve the problems is given afterwards. All this material is part of the introduction of chapter 1, and we finalize it with an outline of the thesis.

Chapter 2 goes into the details of analyzing the architectural structures of Angkor temples. Here we present the core of our discovery. We firstly make a brief discussion on the styles of Angkor temples, and show the relevance of Angkor Wat style temples. We then discuss the successive refinement of the temple complex structure. From the refinement at the element level we select the group of elements with lotus motifs to do further refinements down to the final components which are the lotus motifs themselves. Here we categorize the lotus motifs into different groups and study the rules for structuring them to form a temple element profile. Then we present the mathematical models of the lotus motifs and their properties as well as the mathematical models of the rules and their application.

In chapter 3, we introduce four sets of algorithms, where three of them are for three different methods for reconstructing the 2D element profiles and one for its 3D reconstruction. The first part covers the core of the dissertation in which explicit definitions and conditions are formulated to implement the construction rule. In section 3.2 we present the second method which is designed to alternate the first method in special cases. The third method is based on image segmentation, and is specialized for automatically detecting the 2D element profile from an input image of the element. Finally in the last part of this chapter we show the algorithms for the 3D operation from the 2D profiles.

The results obtained in chapter 2 and chapter 3 will be implemented in the chapter 4. A new software tool called *Angkor Temple Generator (ATG)* which runs parallel with the powerful tool AutoCAD is introduced. A detailed description of ATG architecture and its functionality including possible improvements to the tool is given at the end.

Chapter 5 provides some results on implementing ATG tools to reconstruct Angkor Wat style temples as well as discussing the reconstruction of Khmer temples of other styles.

Finally, in chapter 6 we make our overall conclusions, and describe some possible opportunities for future researches and developments.

Chapter 2

Architecture of Angkor Temple Complex

In this chapter we are going to discuss the architectural structure of Angkor Temples. We begin with defining the temple styles for our framework. The second part will present the successive refinement of the temple complex structure, and discuss in detail on the types of elements especially discussing the types of the lotus motif and their architectural functions. We then try to formulate them in the terms of mathematics.

2.1 Styles of Khmer Temples

Art historian classifies the Khmer temples into fifteen styles with respect to their artistic criteria and the time period they have been constructed. The temple styles are commonly identified by reading the arts of carvings on the *lintel* of the main porch's entrance (see figure 2.1.1) as described in detail in the publications of [Jac99] and [Sti82]. The style of Angkor Wat temple that we have chosen to investigate is defined by the construction period which is between 1113 and 1150 AD during the reign of King *Suryavarman II*.

The reason for the selection of the *Angkor Wat Style Temples (AWSTs)* for our case study is that they surpass the other styles in dimension, complexity and quality of fine arts works. The architecture is exceptionally well and clearly structured. Moreover, if we compare with other styles, the lotus motifs on their elements could also be much easier identified.

Figure 2.1.1
On the left: Lintel of Prasat Kravann. On the right: Lintel of a porch of Angkor Wat Tower.

2.2 Structural Refinement

The aims behind the structural refinement are firstly to better understand the composition of the complex structure, and further to distinguish the types of lotus motifs on each element. Secondly, we need to comprehend and evaluate the geometry of the elements whether they are getting simple enough to implement the CAD 3D operations to model them. If not, further refinement must be done.

In section 2.2.1 we present the refinements of the complex temple structure down to the element level. Then in section 2.2.2 we show the refinement to the element components level which are the lotus motifs. A finite set of lotus motifs are presented with detailed discussion of their architectural functions. Finally, we introduce a rule for structuring the lotus motifs which is the root of the architectural vocabulary of the temple elements.

2.2.1 Elements of the Complex Structure

The structure of Angkor temples can be described as the diagram shown in figure 2.2.1 with an example on Angkor Wat temple structure presented in figure 2.2.2. These figures show that there are three steps of refinement, from complex building level to single building, and two times refinement of the element levels. The refinement of the complex

building at the first step provides the *single buildings* with individual architectural function such as the sanctuary, gopura, gallery, library, etc. The second step of refinement gives two groups of elements: *single elements* and *complex elements.*

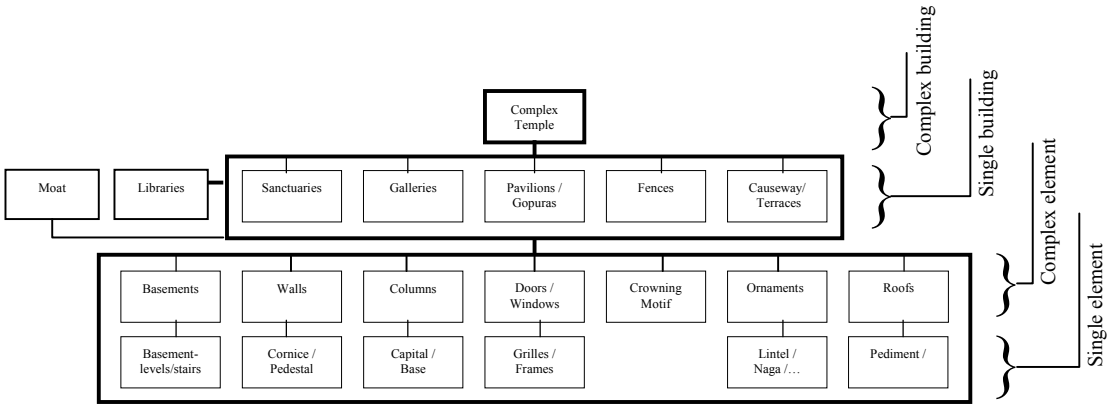


Figure 2.2.1
Diagram of the architectural structure of Angkor Wat style temple.

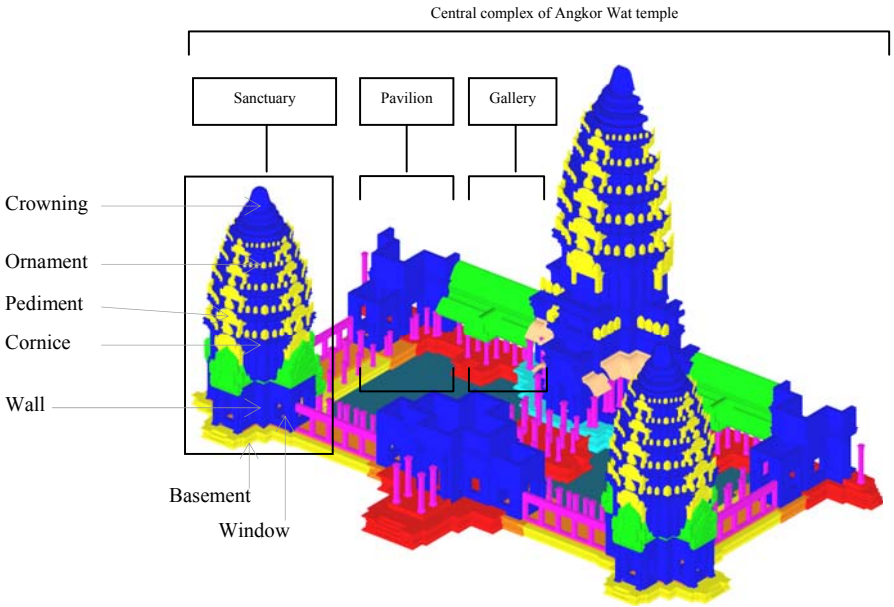


Figure 2.2.2
Example on the 3D model of the central building of Angkor Wat complex.

Definition 2.2.1 (single element). *A temple element, of which the geometry is simple enough to implement the 3D operation to its 2D profile to get the 3D model, is called single element.*

Figure 2.2.3 lists some examples of typical single elements of Angkor Wat style temples. Here we also give a sample for the definition of the single element where a 3D model of a basement is modeled using the 3D operation of extrusion to extrude its 2D profile along an extrude path.

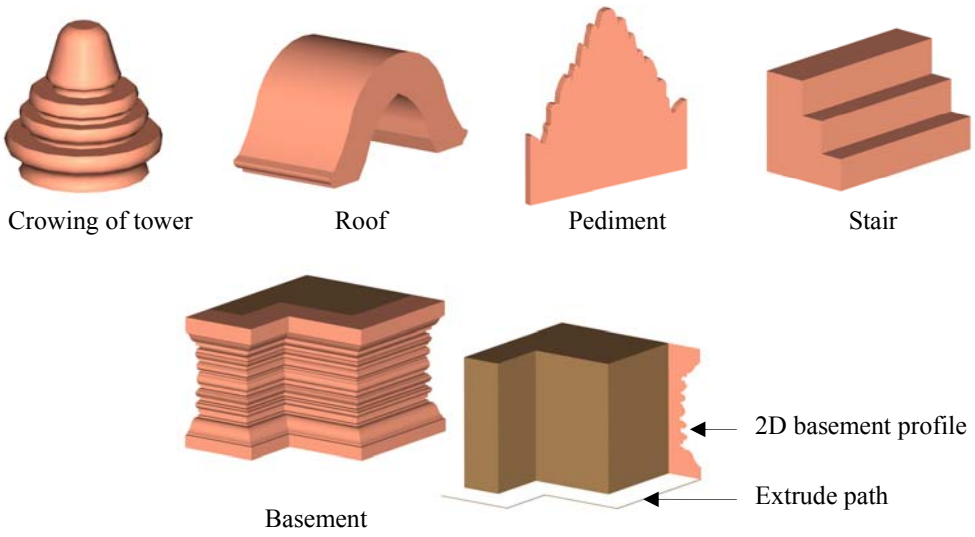


Figure 2.2.3
On top: four samples of typical single elements; below left is a part of a basement 3D model, and on the right is a 2D motif of a basement to be extruded along a given *extrude path* to get the 3D model of a basement.

After the first step of refinement, some groups of elements have not yet become single elements. They are still sets of single elements, and needed to be decomposed. In figure 2.2.4 we list some complex elements, and the way they could be refined into single elements.

Definition 2.2.2 (complex element). *A temple element, which is a composition of more then one single element, is called a complex element.*

Three steps of refinement have been done up to now, and all elements have become single elements. Now let us concentrate on the groups of elements that we would like to investigate, and discussing their relevant properties.

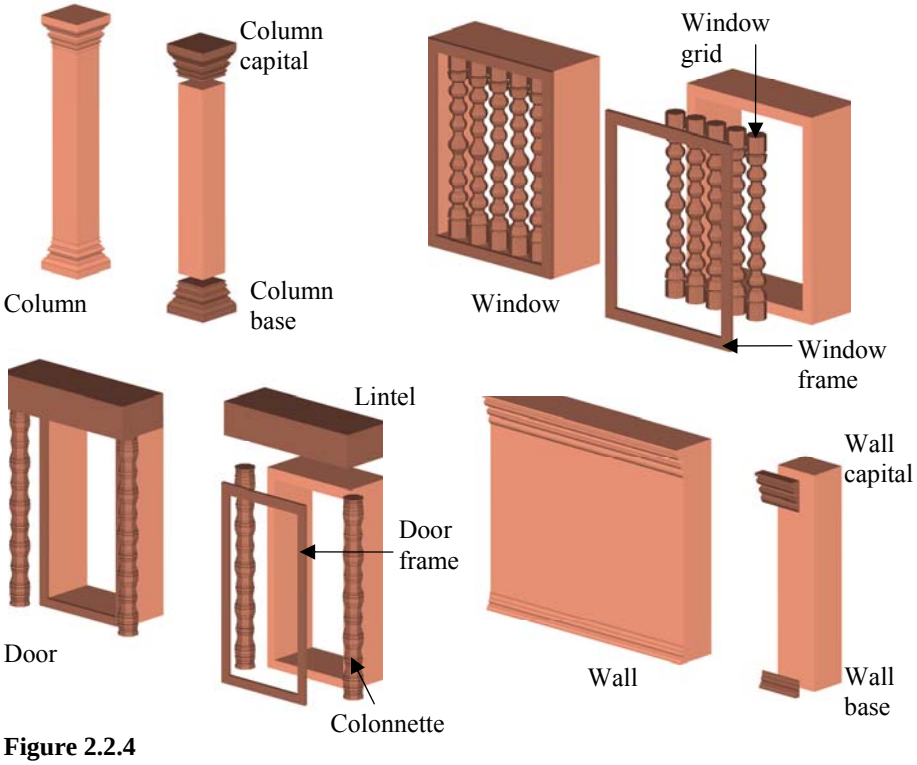


Figure 2.2.4
The refinement of complex elements into single elements.

2.2.2 Modules of Elements

The traditional Khmer art is illustrated in an exceptionally smoothly manner as can be seen from the carvings on almost anywhere of the temple surfaces. Similar is concept for structuring the temple element. They are skillfully combined together in a certain harmonic way.

The left hand side image of figure 2.2.5 shows the side view of a part of a typical Khmer temple porch (3D model is shown in figure 2.2.10) from which we can observe the form of the basement that expresses a smooth visual transition from the flat ground surface to the vertical surface of the basement. Then the transitions from the basement to

the column and the wall are smoothed by the column-base and the wall-base respectively. Also on their tops we can find the column-capital and wall-capital that softened the visual effect of the transition to the adjacent elements.

The curved arrows on the image indicate the described smooth parts. If we compare this temple structure with an identical structure shown on the right hand side which is presented without the moldings that characterize the Khmer temple architecture, we will see the difference. The one on the right clearly shows a sharp transition from one element to another. It can also be interpreted that the architecture of Angkor temple elements provides a steadfast position to the building structure as a whole.

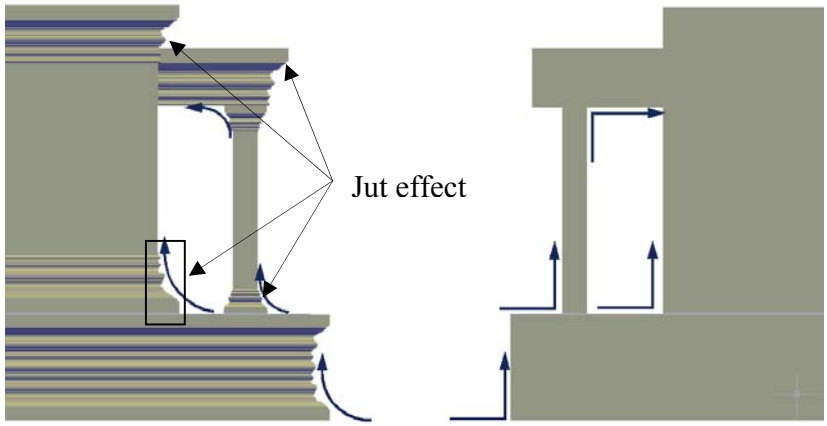


Figure 2.2.5
On the left: Smooth transition from one element to another in Khmer temple architecture.
On the right: Same structure as on the left hand side but without Angkor temple elements.

This architecture develops a so called *Jut effect* at both parts, the *lower part* and the *upper part* of a complex element. We refer these two elements to the *Lower Part Elements (LPE)* and *Upper Part Elements (UPE)* respectively. For instance, a wall consists of a wall-base located at the lower part, and a wall-capital at the upper part of the wall. But this concept is not only specialized for a wall but widely used in Khmer temple complex structure. It is implemented to any element mainly those with a high and smooth vertical surface such as the column, tower tier, temple surrounding fence, etc.

This is one of the reasons why the elements with the decorative molding in form of friezes are decorated almost anywhere on the temple complex.

In terms of art and religious symbolism, these moldings are motifs of lotus petals or lotus flowers depicted in various forms. The lotus in both religions, Hindu and Buddhist, is a holy flower symbolizing the microcosm (individual) of the macrocosm (universe). This is reason for the rich lotus motifs representation on the temple. Figure 2.2.6 gives some examples of their application.

At a closer look at the elements with the molding we see that their structure appear in a unique combination of the lotus motifs which we define as the Angkor Temple Element modules or *ATE modules*.

Figure 2.2.6
Two samples for the lotus motifs dipiction on two elements of Angkor Wat. The left image shows a column-capital of supporting a cornice. Right is a basement of a porch. (Images are done by EFEO).

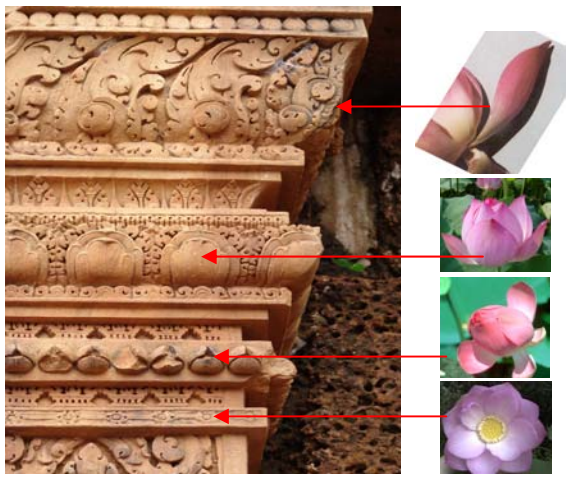


Figure 2.2.7
The left side image shows a column-capital of *Banteay Srey* temple on which four types of lotus motifs are depicted.
The right images show the lotus components and positions corresponding to the motifs on the right image.

Definition 2.2.3 (ATE module). *Temple elements with decorative molding in form of friezes consist of lotus motifs in various forms which are called Angkor Temple Element modules or ATE modules.*

Our investigation of the architectural structure of Angkor temple elements with the molding shows that the number of ATE modules that were used as components is a limited set of six modules. We define them with the characters *a, b, c, d, e* and *f*. In order to better distinguish the type of each module in an element profile we also prefer to present them in six individual colors: magenta for *a*-module type, cyan for *b*, yellow for *c*, green for *d*, blue for *e* and red for *f*. More detailed description of ATE modules and their properties are given in section 2.3.1.

In figure 2.2.8 we give and example of the use of these modules on a basement. Based on these modules we are not only able to create a basement, but also all other ATE elements with the molding including those single elements presented in section 2.2.1. Each element type has its individual properties, particularly the structural composition of its modules. In order to understand the way of composing the modules in a certain type of element we must comprehend the architectural functions of each module. Based on observation and analyzing the structures of various ATEs we can describe the functions of ATE modules as shown in the following section.

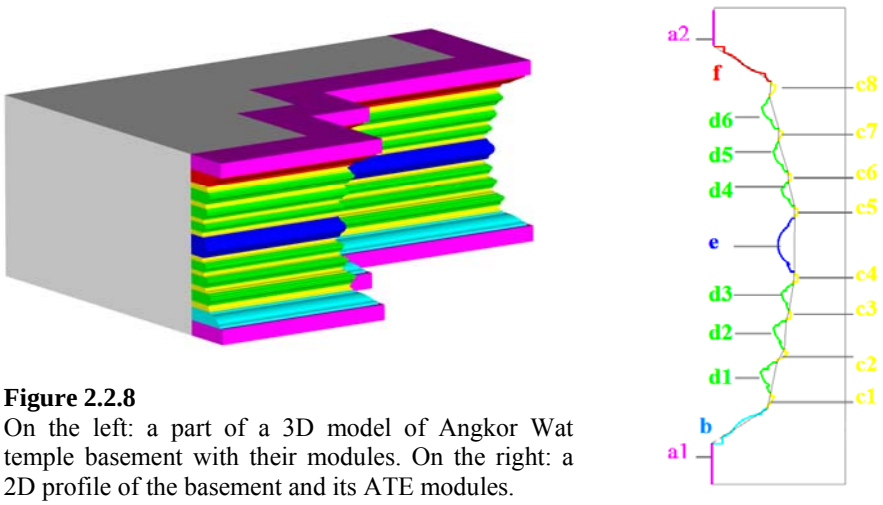


Figure 2.2.8
On the left: a part of a 3D model of Angkor Wat temple basement with their modules. On the right: a 2D profile of the basement and its ATE modules.

2.2.3 Architectural Vocabulary of Elements

Lotus flowers are artistically depicted in various forms but geometrically we can classify these six lotus forms into two different categories: *Convex Class* including *a, b, d, e* and *f,* and *Concave Class c. c*-modules play a significant roll in a module sequence of an element. They are used to link convex modules, and to provide harmonic space between them to bring the convex modules better onto the foreground. We also discovered that modules in the convex class are very rarely directly connected to each other in an element structure. From this architectural setting we know that module *c* must lies between two other convex modules. Though exceptional cases occur in linking *a* with *b* and *f* with *a*.

Module *b* and *f* are responsible for providing a aesthetic jut effect of an element as a whole. This can be seen in their geometrical structure that shows the relatively small and high angle value of the vectors of *b* and *f* respectively in comparison with any other modules (see section 2.3.1). The geometry of *f* is identical to the geometry of a reflection of *b* by a horizontal line, and both of them do not exist without any possible links with *a*, where the existence of *a* is independent. We can summarize the relationship of *b* and *f* with *a* as a formation of a "(*ab*) and (*fa*) block". On the other hand, *a* is a module which can be considered as the sequence *Initiator* and *Terminator*. *a* mostly appear twice, found at both the ends of an element sequence. However, it sometimes can also appear in between two other modules on the basement element. In this case, module *a* replaces *e* exactly at the central part of the sequence. We should make notice at this point that the highest number of *a* in an ATE module sequence is three, and is only found on a basement. Otherwise *a* appears once as the initiator in the case of the wall-capital of Angkor Wat temple.

Module *e*, as described at the beginning, is a convex module, and is always linked with *c* to both sides of itself. A considerable property of *e* is its varying prosition in the sequence up to the type of element. Related to a basement for instance is the position of *e* in the middle of the sequence whereas on a wall-bas or column-base it is at the third last position if counting the modules in that sequence bottom-up. The other way round exists

e always at the third range on the wall-capital or column-capital. Basically an *e*-module exists only once in an ATE. An exception, however, is found on the *Banteay Srey*. On the basement of Banteay Srey we can find the *e*-module twice in the sequence. This is a very special case, and is not found on Angkor Wat style temples.

Now let us get back to discuss the last module *d*. Module *d* is probably the most important module amongst the five convex modules. It is what we would call the *Repeated Module*, and responsible for the complexity of the module sequence. Thus an element is more or less complex depending on the number of *d*-modules. *d* and *e* belong to the convex class modules with the most basic property of combination because they are allowed to link at their both ends only with *c*. Elements with composition of multiple *d* constitute (similar case to (*a*, *b*) and (*f*, *a*) but more complex) a "(*cd*) block". Figure 2.2.8, for example, shows such kind of (*cd*) block in a basement where the block consisting of $6 \times d$ separated into two blocks, the upper and the lower (*cd*) block by an *e*-module. This separation is a typical characteristic for the basement element. However, in other element types the (*cd*) block remains being a single continuous (*cd*) block. Such cases occur to wall-base, wall-capital, column-base, column-capital, crown etc. Later, in chapter 3, we will see the advantages of this architecture for our reconstruction algorithm.

At this point we would like to recall some remarks we have made in section 2.2.2 about the properties of the lower part elements and upper part elements of a complex element such as the wall. If we have a closer look at their module structures, we can identify the relationship between the structural composition of these two elements and the structure of a basement. Figure 2.2.9 shows how to transform a sequence of a basement into a wall-base and a wall-capital or the other way round. At the middle part (*central module*) of the basement sequence, which is in this case the *e*-module, we separate the sequence into two parts in such a way that after separating the sequence each part get an *e*-module at the separation point. Then at that points we add (c_4, a_2) and (a_1, c_1) to the lower part element and the upper part element respectively. We now obtain two new proper sequences, a wall-base and a wall-capital.

The 2D geometric profile of the wall-base and wall-capital appear to be the reflection of one another with a horizontal mirror-axis. It is true if we considering the range of the module in the sequence, but geometrically they are not the reflection of one another. Figure 2.2.10a and 2.2.10b shows that module *a* and *e* do not change their appearance because they are symmetric whereas *f* is identical to a horizontal flipping of *b*-module. But *d* and *c* appear abnormal. In this context, it is not possible to reconstruct a lower part element, and mirror it in order to get the upper part, and the other way round.

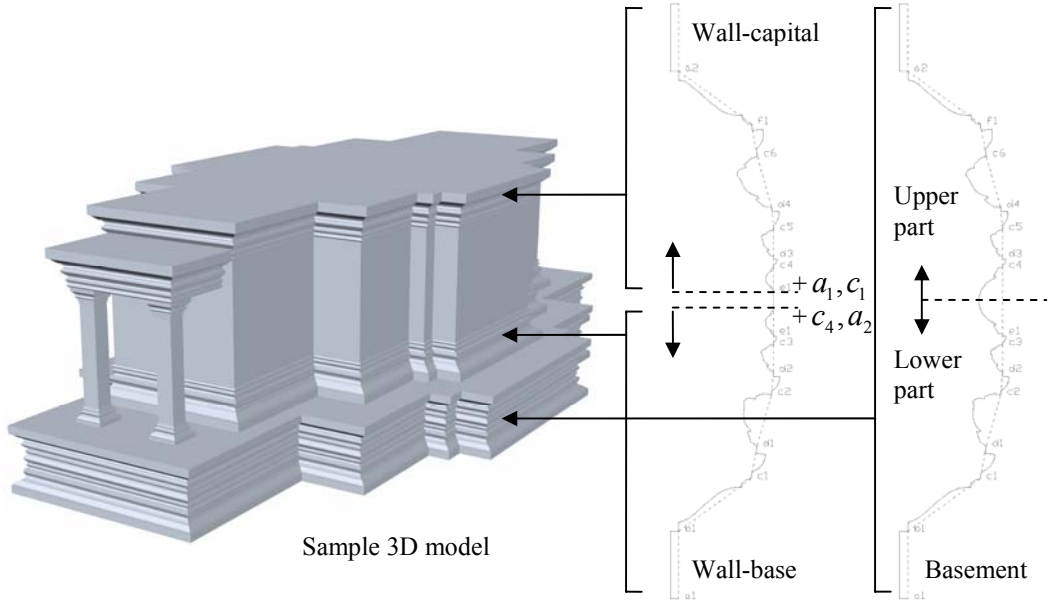


Figure 2.2.9
On the left: Sample 3D model of a typical simple Khmer temple architecture in which the structure of the wall-base and wall-capital are products of a separation of its basement structure. On the right: 2D profile of the three related elements.

Figure 2.2.10a
the wall-base and wall-capital of Chao Say Tevda temple in Angkor.

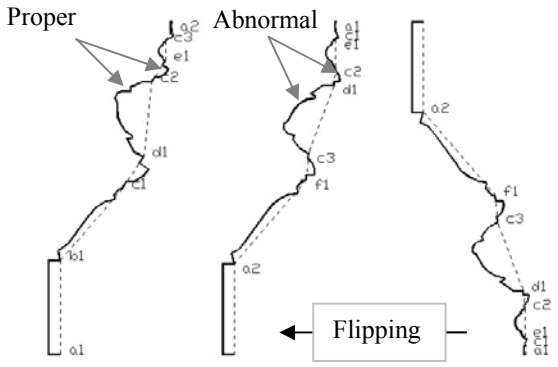


Figure 2.2.10b
Lower 3 figures are 2D profile of:
Left: wall-base,
Middle: vertical flip of the wall-capital,
Right: wall-capital.

Let us consider the modules as line segments that connect the start point with the end point of each module (figure 2.2.10b presented them as dashed line segments). Their start points and end points are defined by the nodes (*s*) and node (*e*) respectively. The line segment is the vector that represents the type and the proportion of the corresponding module. We define a bottom-up reconstruction procedure of the module sequence, thus representing the vector as a vertical line segment with the end point (*e*) at the top of line.

Based on the analysis of the architectural function of ATE modules in Khmer temples, we can describe the rule for the connectivity of the modules in from of a graph as shown on left hand side of figure 2.2.11. The dashed lines here identify the allowed links of two different types of modules. Suppose we want to reconstruct the sequence *S* of the wall-base shown in figure 2.2.10b with the input modules $S_{wall-base} := (2a, b, 3c, d, e)$. We know that the sequence begins with an *a*, in this case a_1. Then, even though a link between *a* and c is allowed, the connection, green colored arrow number 1 pointing from the end point of a_1 to the start point of b has the priority, see the right hand side image of figure 2.2.11.

The next module must be a c_1 since there is no other possible links with the end point of b. From c we have four possible links: to *a*, *d*, *e*, and to *f* however, *f* is not available, and a_2 must be at the end of the sequence. Module *e,* as mentioned earlier, is used as the third last convex module in a wall-base element which is not the current case.

Thus the last choice is *d*. Then follows c_2, e, c_3 and finally a_2. As a result, we get a unique sequence of modules:

$$S_{wall-base} = (a_1, b, c_1, d, c_2, e, c_3, a_2).$$

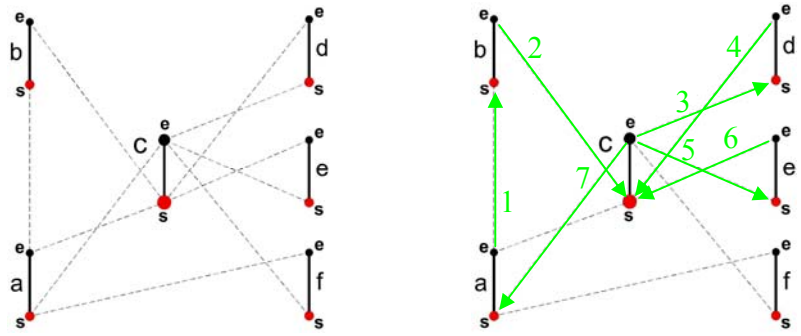


Figure 2.2.11
Left: principle rule of structuring the ATE modules. Right: sample for connecting modules stepwise from 1 to 7 to reconstruct the wall-base sequence shown in figure 2.2.10b.

The principle rule we have set up is responsible for restricting the connectivity of modules in order to prevent improper reconstruction of ATE module sequence. Nevertheless, in some cases there is an ambiguity where more then one proper sequences can be reconstructed using one single set of input modules. For instance, both sequences below, sequence (1) (for a basement) and (2) (for a wall-base) have the same set of input modules $S := (2a, 4c, 2d, e)$. Moreover, we observe that the convex modules *a*, *d* and *e*, have the same connectivity to *c*-module. Suppose there is no restricted priority of using the three modules. In this case we would obtain several improper sequences when using the set of input modules as shown in figure 2.2.10 as an example. The results are presented in figure 2.2.12 where *P(S)* denotes the final element profile of which *S* is the imbedded module sequence.

$$S_{basement} = (a_1, c_1, d_1, c_2, e, c_3, d_2, c_4, a_2) \qquad (1)$$

$$S_{wall-base} = (a_1, c_1, d_1, c_2, d_2, c_3, e, c_4, a_2) \qquad (2)$$

$$S_1 = (e, c_1, d, c_2, a_1, b, c_3, a_2)$$

$$S_2 = (a_1, c_1, a_2, b, c_2, d, c_3, e)$$

$$S_3 = (d, c_1, e, c_2, a_1, b, c_3, a_2)$$

$$\dots$$

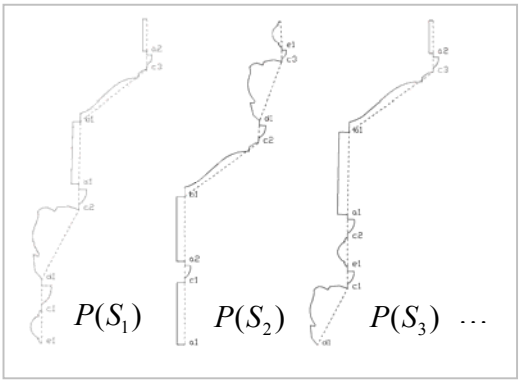


Figure 2.2.12
Samples for accidental reconstructions of module sequences using the principal rule without restriction to the priorities of *a, d* and *e*-module.

It is therefore very important to take into account the assigned priorities and key roles of the modules for a certain element type in order to ensure a proper reconstruction. Implementing the principal reconstruction rule we are able to avoid critical errors but it does not provide sufficient conditions to solely implement this rule for reconstructing any type of Angkor temple element without additional information of the element itself. For this reason it is necessary to discuss the properties of all elements in relation to their components individually in detail. We will do this mathematically in the next section.

Angkor temples, particularly the Angkor Wat style temples are unique in their architecture and complexity. Their specific architectural concepts allow us to decompose the whole complex structure into simpler components at multiple levels. This process on the other hand provides great advantages not only for better understanding the complex structure as well as the structure of each element but also opens a new way in terms of 3D computer reconstruction. With this knowledge we were able to setup a new rule for reconstruction of the most complex elements using a module-based reconstruction method. In the next section we are going to describe all material we have presented here in a more advanced and mathematical way.

2.3. Abstract Geometry of Temple Elements

From the previous section we know that Angkor temple elements are composed of a finite set of modules that allow us to flexibly reconstruct various forms of element profiles. Before we describe an element structure, we need to know some initial properties of each module including position, proportion and orientation. In section 2.3.1 we describe the modules as abstract objects with all their detailed properties and express how they are presented in a sequence. Then we present the abstract geometry of each *module profile*. The last section 2.3.2 of this chapter is about mathematical modeling of a sequence structure and its relevant properties such as the position and proportion of each module.

2.3.1 Modules as Abstract Objects

An ATE module graphically is a series of lines and arcs segments, each of which consists of specific properties, forming a certain non-closed sequence of line and arc segments called *module profile* (see definition 2.3.2). Module profiles have two ends, the *start point* and the *end point*. These two nodes together define an edge that we call *module vector* which is responsible for the proportion, direction and position of the corresponding module profile in a module sequence of ATE. Thus we should keep in mind that the main subject of reconstruction an Angkor temple element profile is dealing with arrangement of the given module vectors and their proportional and directional settings.

Let t be the module vector of type $t \in \{a,b,c,d,e,f\}$, and let $s^t = (x_s, y_s)$, $e^t = (x_e, y_e) \in \mathbb{R}^2$ be the start point and the end point of the vector represented by the edge $t(s,e)$ respectively. The distance from the starting point s^t to the end point e^t of the module is defined by $l_t \in \mathbb{R}^{>0}$. In addition let $h_0 := (x_s + 1, y_s)$ and define $\alpha_t(s^t, h_0, e^t)$

for the size of the angle measured in radian so that t can be rotated in counter-clockwise direction over the starting point s, with $0 < \alpha < \pi$.

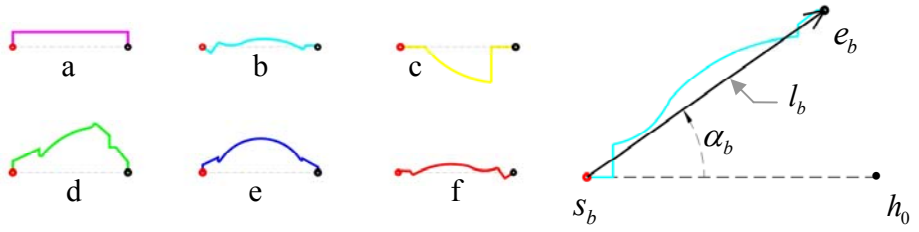


Figure 2.3.1
On the left: six basic ATE modules, *a, b, c, d, e,* and *f* presented at start position. On the right: rotating the module type *b* by $\alpha_b(s^b, h_0, e^b)$ in counter-clockwise over the start point s^b.

The very first step of reconstructing a sequence of ATE module vector is initializing each module vector t at the start position, without changing direction and proportion.

Definition 2.3.1 (start position)
A module $t(s,e)$ *with start point s and end point e is said to be at the start position if it is initialized at the origin of the coordinate system thus* $s^{t_0} = (0,0)$, *and a magnitude and the angle* $l_{t_0} = 1$, $\alpha_0 = 0$ *respectively.*

Before we discuss on the model of a module structure and temple element profile representation we would like to present the abstract model of the six ATE module profile and required parameters for their generation.

Module profiles are two dimensional curves consisting of *n* line segments and *m* arc segments connecting to each other forming a specific motive representing one of the six lotus motives. Let us define $p_t(s,e)$ to be the module profile of type *t* with $\lambda(\sigma)$ line segments and $\mu(\sigma)$ arc segments with $\{\lambda, \mu \in p_t;\ \sigma_{i,t}(x_i, y_i) \in \mathbb{R}^2 \quad | \forall i = 1,...,n\}$, in which $M_i(x,y) \in \mathbb{R}^2$ be the central point, and $r_j \in \mathbb{R}^{>0}$ be the radius of arc μ_j for all $t \in \{a,b,c,d,e,f\}$. Corresponding to [AG89] and [Pap01], the profile $p_t(s,e)$ can be described by a function:

$$f_i(x,y)=\begin{cases} f_\lambda=(x_{i+1}-x_i)(y-y_i)-(y_{i+1}-y_i)(x-x_i)=0 & (2.1)\\ f_\mu=y_{M_i}-y+A=0 \text{ with } A=\pm\sqrt{r_i^2-(x-x_{M_i})^2}\in\mathbb{R} & (2.2)\end{cases}$$

such that if $f_i(x,y)$ describes a line segment $\lambda=(\sigma_i,\sigma_{i+1})$ then we implement the function (2.1). Else if it describes an arc segment $\mu=(\sigma_i,\sigma_{i+1})$ then we implement the function (2.2) with $A=-\sqrt{r_i^2-(x-x_{M_i})^2}$ if $\sigma\in\mu\le y_M$, and $A=\sqrt{r_i^2-(x-x_{M_i})^2}$ if $\sigma\in\mu>y_M$. Finally, we define the start point and the end point of the profile as $s^p=\sigma_1$ and $e^p=\sigma_n$ respectively.

Definition 2.3.2 (module profile)

A module profile p_t is a two dimensional sequence of non-closed $n-1$ line segments $\lambda(\sigma)$ and arc segments $\mu(\sigma)$, connected to each other by n nodes with $\{\lambda,\mu\in p_t;\ \sigma_{i,t}(x_i,y_i)\in\mathbb{R}^2\,|\,\forall i=1,...,n\}$ each of which can be analytically described by $f_{i,\lambda}(x,y)\Rightarrow(2.1)$ and $f_{i,\mu}(x,y)\Rightarrow(2.2)$ where $s^{p_t}=\sigma_1$ be the start point and $e^{p_t}=\sigma_n$ the end point of p_t.

As we can see in figure 2.3.1 that the module profile of type *a* has the simplest curve amongst the other five. It consists of only three line segments thus $\sigma=4$, $\lambda=3,$ and $\mu=0$. The ideal profile p_a is initialized at the start position using following input parameters to describe it:

$$\sigma_1=(0,0),\ \sigma_2=\left(0,\frac{3}{25}\right),\ \sigma_3=\left(1,\frac{3}{25}\right),\ \sigma_4=(1,0).$$

We obtain the equations of each line by using the function $f_\lambda(x,y)$:

$$f_1=(0-0)(y-0)-\left(\frac{3}{25}-0\right)(x-0)=-x\left(\frac{3}{25}\right)=0;\ \Rightarrow f_1=x=0$$

$$f_2 = (1-0)\left(y-\frac{3}{25}\right)-\left(\frac{3}{25}-\frac{3}{25}\right)(x-0) = y-\frac{3}{25} = 0;\ \Rightarrow f_2 = y = \frac{3}{25}$$

$$f_3 = (1-1)\left(y-\frac{3}{25}\right)-\left(0-\frac{3}{25}\right)(x-1) = \frac{3}{25}(x-1) = 0;\ \Rightarrow f_3 = x = 1.$$

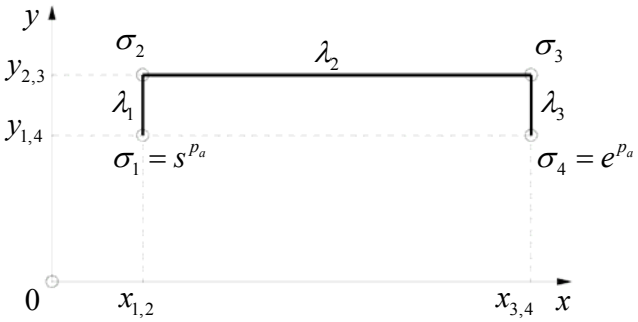


Figure 2.3.2 graphics representtation of the curve of *a*-module profile p_a (moved from the origin).

Other type of module profile such as module *b* is slightly more complex. It is composed of five lines: $\lambda(\sigma_1,\sigma_2),\lambda(\sigma_3,\sigma_4),\lambda(\sigma_5,\sigma_6),\lambda(\sigma_6,\sigma_7),\lambda(\sigma_7,\sigma_8)$ and two arcs: $\mu(\sigma_2,\sigma_3)$ and $\mu(\sigma_4,\sigma_5)$. Thus p_b must be described by switching between both functions corresponding to their belonging *i* index shown as following:

$$p_b := \left\{ \begin{array}{l} f_\gamma(\sigma_i,\sigma_{i+1}) \text{ with } i \in \{1,3,5,6,7\} \\ f_\mu(\sigma_i,\sigma_{i+1}) \text{ with } i \in \{2,4\} \text{ such that } \left| \begin{array}{l} A = -\sqrt{r_i^2-(x-x_{M_i})^2} \text{ if } i = 2 \\ A = \sqrt{r_i^2-(x-x_{M_i})^2} \text{ if } i = 4 \end{array} \right. \end{array} \right\}.$$

The graphics of an ideal p_b with parameters of each node we have chosen for modeling it in *CAD* tool is shown in figure 2.3.3.

Similarly, the other modules *c*, *d*, *e*, *f* are comprised of both line and arc segments. They can be described using the same equations and the corresponding conditions but with different properties including number of arc- and line segments and their input

parameters. Figure 2.3.4a and 2.3.4b list the input parameters for modeling their geometry.

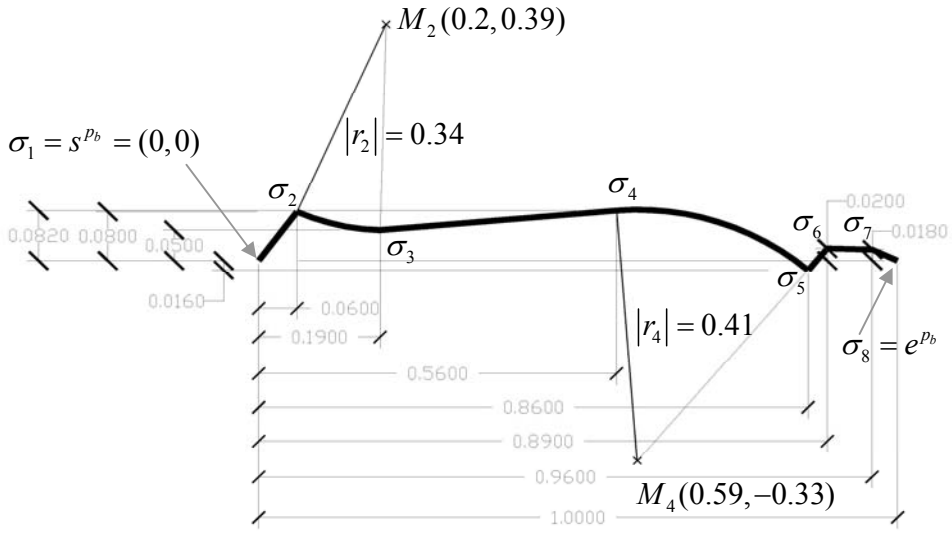
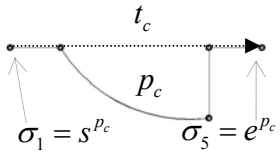
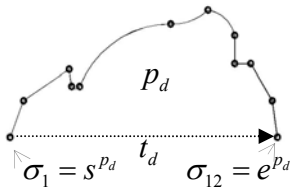


Figure 2.3.3
Graphics representation of module profile p_b and initializing at the start position with its measurements.

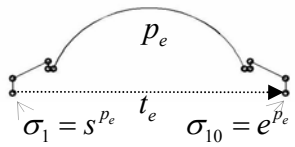


p_c	i	σ_i		σ_{i+1}		r	$\sqrt{r^2 - (x - x_M)^2}$
		x_i	y_i	x_{i+1}	y_{i+1}		
Line	1	0	0	0.2	0		
Arc	2	0.2	0	0.79	0.29	0.6	(-)
Line	3	0.79	0.29	0.79	0		
Line	4	0.79	0	1	0		

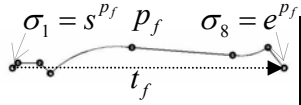
Figure 2.3.4a
Graphs and tables of parameters for p_c initialized at the start position.



p_d	i	σ_i		σ_{i+1}		r	$\sqrt{r^2-(x-x_M)^2}$
		x_i	y_i	x_{i+1}	y_{i+1}		
Line	1	0	0	0.5	0.13		
Line	2	0.5	0.13	0.22	0.24		
Line	3	0.22	0.24	0.23	0.18		
Arc	4	0.23	0.18	0.26	0.18	0.19	(-)
Arc	5	0.26	0.18	0.6	0.4	0.36	(+)
Arc	6	0.6	0.4	0.74	0.45	0.19	(-)
Arc	7	0.74	0.45	0.84	0.36	0.12	(+)
Line	8	0.84	0.36	0.84	0.26		
Line	9	0.84	0.26	0.9	0.26		
Line	10	0.9	0.26	0.98	0.13		
Line	11	0.98	0.13	1	0		



p_e	i	σ_i		σ_{i+1}		r	$\sqrt{r^2-(x-x_M)^2}$
		x_i	y_i	x_{i+1}	y_{i+1}		
Line	1	0	0	0	0.06		
Line	2	0	0.06	0.13	0.13		
Line	3	0.13	0.13	0.13	0.1		
Arc	4	0.13	0.1	0.15	0.1	0.13	(-)
Arc	5	0.15	0.1	0.85	0.1	0.37	(+)
Arc	6	0.85	0.1	0.87	0.1	0.13	(-)
Line	7	0.87	0.1	0.87	0.13		
Line	8	0.87	0.13	1	0.06		
Line	9	1	0.06	1	0		



p_f	i	σ_i		σ_{i+1}		r	$\sqrt{r^2-(x-x_M)^2}$
		x_i	y_i	x_{i+1}	y_{i+1}		
Line	1	0	0	0	0.02		
Line	2	0	0.02	0.1	0.02		
Line	3	0.1	0.02	0.14	-0.2		
Arc	4	0.14	-0.2	0.44	0.08	0.33	(+)
Line	5	0.44	0.08	0.81	0.05		
Arc	6	0.81	0.1	0.94	0.08	0.17	(-)
Line	7	0.94	0.08	1	0		

Figure 2.3.4b
Graphs and tables of parameters for forming the geometry of p_d, p_e and p_f module profile initialized at the start position.

Now let us turn back to the module vectors. The Reconstruction of ATE sequences of module vectors in an initialized state gives a sequence of *N* homogeneous line segments forming up a horizontal line as a whole. By providing a certain individual value of length and angle to each module vector we obtain a specific shape of ATE profile. The operations and analysis of proportion and direction of module vectors will be discussed in the next chapter. By now we need to define the mathematical model and the graphics representation of the module structure (see also figure 2.3.5).

Definition 2.3.5 (ideal ATE profile)

A module sequence $S_{ATE} = (t_1,...,t_i,...,t_N)$ *of an ATE with* $t_i(l_i,\alpha_i) := \{l_i \in \mathbb{R}^{>0}, \alpha_i = (s^{t_i}, h_0, e^{t_i}) \le \pi\}$ *and the element profiles* $P := \{p_{i,t} \mid \forall i = 1,...,N\}$ *such that* $p_{i,t} \in P \mapsto t_i \in S_{ATE}$ *in such a way that* $s^{p_{i,t}} = s^{t_i}$ *and* $e^{p_{i,t}} = e^{t_i}$ *for* $t \in \{a,b,c,d,e,f\}$*, is called an ideal ATE profile, and is described as* $P_{ATE} = (p_1,...,p_i,...,p_N)$.

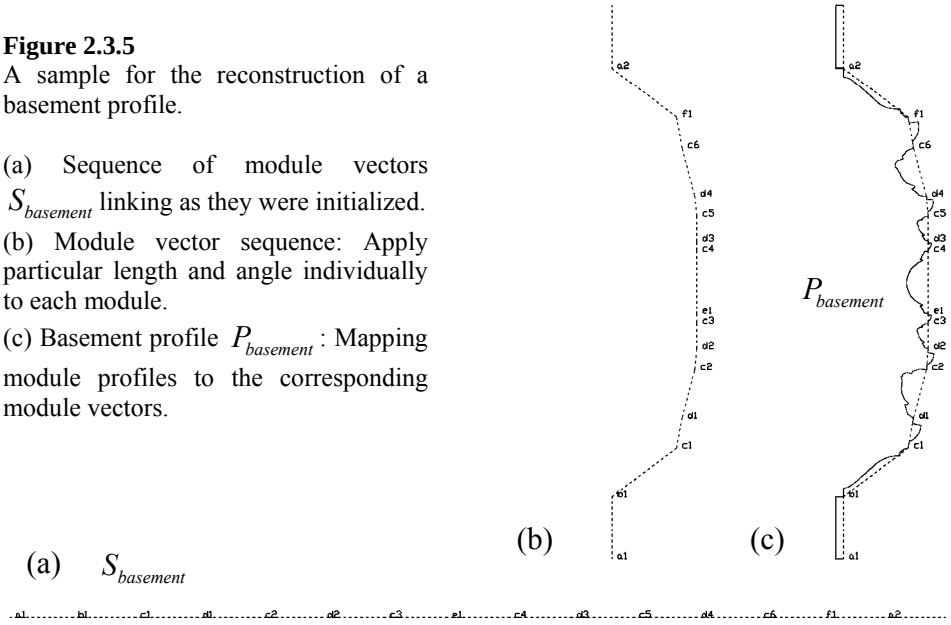


Figure 2.3.5
A sample for the reconstruction of a basement profile.

(a) Sequence of module vectors $S_{basement}$ linking as they were initialized.
(b) Module vector sequence: Apply particular length and angle individually to each module.
(c) Basement profile $P_{basement}$: Mapping module profiles to the corresponding module vectors.

When reconstructing an element profile, especially the reconstruction of an existing element, we sometimes need to modify the predefined ideal properties of the element structure in order to fit to a particular requirement. It is therefore necessary to have access to any components in the module sequence and its parameters.

In the data structure of an element profile, a module t_j is identified by the *position* of its start point s^{t_j} for all $i = 1,..., j$ and $i < j < N$. From the definition of module sequence of ATE we know that the modules consist of individual length and angle, and are linked to each other as a chain. This structuring approach causes s^{t_j} to depend on the properties of the preceding module t_i. The position of t_j is therefore defined and is calculated as shown in definition 2.3.6 (see also figure 2.3.6).

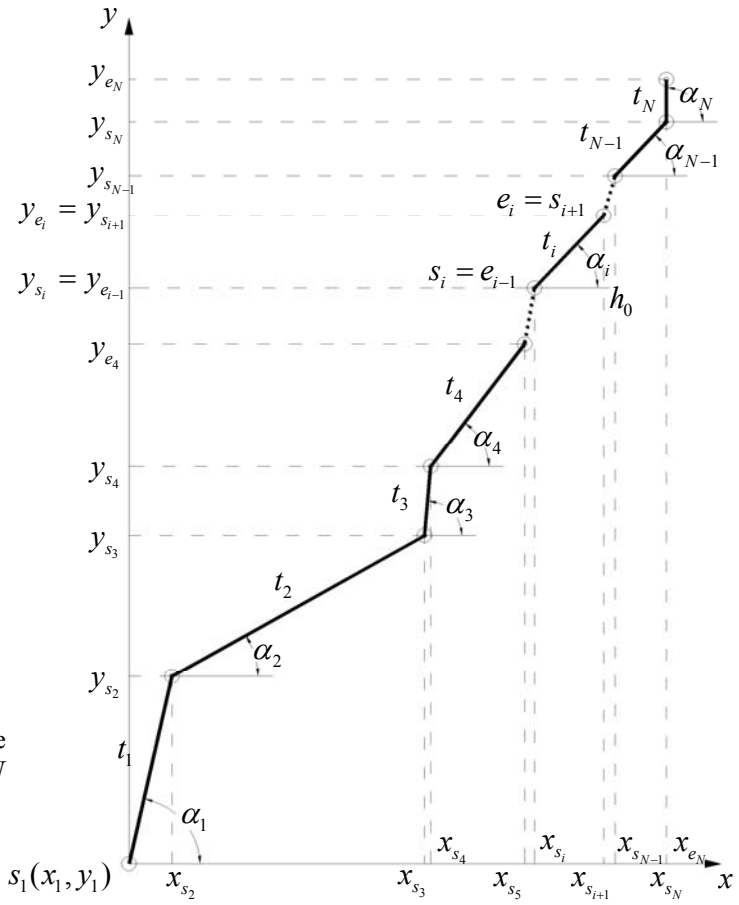


Figure 2.3.6
A graph of an LPE module sequence structure with N input modules.

Definition 2.3.6 (position)

Let l_i *and* α_i *be, respectively, the length and the angle of* $t_i(s,e) \in S_{ATE}$ *for all* $i = 1,...,N$. *The position of a module* t_j *with* $i < j < N$ *is defined by the coordinate of its start point* s^{t_j}, *and can be calculated as following:*

$$s^{t_j}(x_j, y_j) := \left\{ x_j = \sum_{i=1}^{j} l_i \cdot \cos\alpha_i,\ y_j = \sum_{i=1}^{j} l_i \cdot \sin\alpha_i \right\}.$$

Another important property of P_{ATE} is its *ideal height*. As the individual proportion (length and angle) is given to each module in an element module sequence, we obtain a 2D element profile with an ideal height (see definition 2.3.5). We need to know this height in order to compute the required height of the element. We will present the scaling transformation in detail in chapter 3. For the moment we need to give a definition of the ideal height of *P*.

Let $h \in \mathbb{R}^{\geq 0}$ be the *ideal height* of the ideal ATE profile P_{ATE}. We define and calculate the ideal height as following:

Definition 2.3.7 (ideal height)

The difference $y_{e^{t_N}} - y_{s^{t_1}}$ *of the end point* e^{t_N} *of the last module and the start point* s^{t_1} *of the initial module in an ATE profile* $P_{ATE} := \{S = (t_1,...,t_i,...,t_N) \text{ with } t_i = t(l,\alpha) | \forall i = 1,...,N\}$ *is defined as the ideal height h of the ideal ATE profile* P_{ATE}: $h = y_{e^{t_N}} - y_{s^{t_1}}$

2.3.2 Abstract Geometry of Module Sequence

We have seen that the reconstruction of an ATE module sequence is based on the principle rule that we have presented in section 2.2.3. We also have provided some introduction on the implementation of the rule. In this section we will describe the rule and the implementation more mathematically.

The module connectivity as shown in figure 2.2.11 describes a restriction of the combinatorial problem where only the end point of a module t_1 is allowed to the start point of the proceeding module t_2. Furthermore, only module types which are directly connected with dashed lines are allowed to link to each other. A network with such properties can be represented by a *directed graph* or *digraph G* shown in figure 2.3.7.

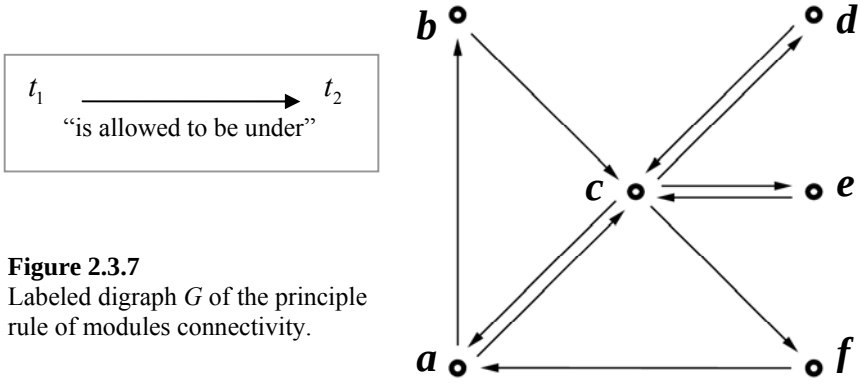


Figure 2.3.7
Labeled digraph *G* of the principle rule of modules connectivity.

Definition 2.3.3 (digraph $G=(V,E)$)

Let V be a set of six vertices $v_t \in V$ labeled by $t \in \{a,b,c,d,e,f\}$ and let E be a set of ten directed edges (arrows) $\varepsilon \in E$ each of which composing of two adjacent vertices $\{v_{i,t}, v_{i+1,t}\}$. The direction of the edge $\varepsilon = [v_{i,t}, v_{i+1,t}]$ defines the allowed link between v_i and v_{i+1} thus between two different module types (t_i) and (t_{i+1}). The digraph G is described as:

$$G=(V,E)=\left(\{v_a,v_b,v_c,v_d,v_e,v_f\},\left\{\begin{matrix}[v_a,v_b],[v_a,v_c],[v_b,v_c],[v_c,v_a],[v_c,v_d],\\ [v_c,v_e],[v_c,v_f],[v_d,v_c],[v_e,v_c],[v_f,v_a]\end{matrix}\right\}\right)$$

The digraph *G* has an *order* of six; $|G|=6$ representing the six module types *t,* and consisting of ten edges (arrows); $\|G\|=10$. The vertex v_c has the highest *degree*

$d(v_c) = 4d_{in}(v_c) + 4d_{out}(v_c) = 8$ denoted by the connectivity to v_a, v_d, v_e and v_f where $d_{in}(v_t)$ (*in-degree*) denotes the arrow pointing to v_t, and $d_{out}(v_t)$ (*out-degree*) denotes the arrow pointing out from v_t. Then it follows by $d(v_a) = 2d_{in}(v_a) + 2d_{out}(v_a) = 4$, and the rest four vertices each with degree of $d_{in}(v_t) + d_{out}(v_t) = 2$ for $t \in \{b,d,e,f\}$, [Die01].

In the definition of graph G we defined the label of each $v_t \in G$ by $t \in \{a,b,c,d,e,f\}$, and in addition we define $d_{in}(v_t)$ and $d_{out}(v_t)$ to be the connectivity of the start point s^t and the end point e^t of t respectively. Thus we must keep in mind that instead of referring $t(s^t, e^t)$ to the edge $\varepsilon \in G$, it is referred to the vertex $v_t \in G$ such that the type of $t \to v_t$ with $s^t = d_{in}(v_t)$, $e^t = d_{out}(v_t)$ and $t \in \{a,b,c,d,e,f\}$. The graph G can now represent the network of connectivity of two different types and adjacent modules t_i and t_{i+1} in an ATE module sequence $S_{ATE} = (t_1,...,t_i,...,t_N)$.

Based on [PS82] and [NR04], a *directed walk* $W = (V', E')$ of G, consisting of a pair sets of vertices V' and edges E', is generally defined as a finite non-null sequence of vertices $W = (v_1,...,v_i,...,v_N)$, $N \geq 1$, such that $e_j = [v_j, v_{j+1}] \in E'$ for $j = 1,...,N-1$. A directed walk $W_N(G)$ along N vertices $v \in G$ from v_i to v_{i+1} is restricted to walk in a fixed direction of the edge $\varepsilon = [v_i, v_{i+1}]$. Thus we define W_N is equivalent to the formation of the module sequence S_{ATE} with N number of input modules $t_i(s_i, e_i) \in S_{ATE}$ $\forall i = 1,...,N$ in which the start point $s^{t_{i+1}}$ of the successor module t_{i+1} joins with the end point e^{t_i} of the preceding module t_i, thus $s^{t_{i+1}} = e^{t_i}$.

Definition 2.3.4 (module sequence of ATE)

A module sequence S_{ATE} of an Angkor Temple Element that is composing of N modules of type t is equivalent to the sequence of the walk W along the vertices $\{v_i \in W_N(G) | \forall i = 1,...,N\}$ and is described as:

$$\{S_{ATE} = (t_1, ..., t_i, ..., t_N)\} \equiv \{W_N(G) = (v_1, ..., v_i, ..., v_N)\} \text{ with } s^{t_{i+1}} = e^{t_i} \text{ and } t_i \to v_i .$$

The section 2.2.3 shows some examples for typical module sequence structures of ATE. We can see that modules of the same type can occur multiple in a sequence. This means that the walk W is allowed to pass through a certain vertex $v_t \in G$ multiple times. In such cases, W is not a *path* however, in simple structured sequences, e.g. $S_{wall-capital} = (a, c, d)$, where a vertex $v_t \in G$ is passed only once, W can be considered as a path. On the other hand, we know that generally the sequence S_{ATE} is initialized and terminated with module type $t_1 = t_N = a$. This property makes W to be a *closed walk* or a *cycle* with $v_1 = v_N$ in almost any case of reconstructing the ATE module sequence. Therefore W is for our purpose not fixed to a specific type of walk. Whether it is a path or a cycle or even none of these both kinds, it definitely depends on the composition for certain Angkor temple element architecture.

So far, we have set a mathematical model of the principle rule for reconstruction of ATE module sequence, and we know its implementation to obtain a proper sequence. Based on this knowledge and the models we have introduced in this chapter, we proceed to discuss on the algorithms for reconstruction of module sequences of Angkor temple elements in the following chapter.

Chapter 3

Algorithm for Reconstruction of Angkor Temple Elements

In this chapter we will present the algorithms for the 3D reconstruction of Angkor temple elements. The first three sections discuss about three different algorithmic approaches for the reconstruction of the 2D element profile, and in the last section we discuss the algorithm for the 3D reconstruction of the temple element based on the 2D profile that we have obtained from the previous sections.

The first section of this chapter covers the discussion of the algorithms for the reconstruction of the 2D element profile based on the predefined modules and the construction rule we have presented in chapter 2. In the section 3.2 we will present another algorithm for the reconstruction of the 2D profile which is also based on the six predefined modules but the sequence and proportions of the modules are manually captured from an input image. The section 3.3 is about the third reconstruction algorithm which is apart from any predefined objects and rules. This algorithm is based on *Image Segmentation* where the element profile in an input image is automatically detected.

In the last section of this chapter we discuss about the 3D reconstruction of Angkor temple elements. Here we will show the essential operations to convert the 2D profile into the 3D model.

3.1 Reconstruction Based on Number of Modules

The core of our research will be presented in this section. Most of the knowledge we have gained from studying the architectural structure of Angkor temple is applied in this first part. We will begin with setting up a general algorithm and the required conditions for using the construction rules. Afterwards, we discuss the reconstruction of particular element structures which require specific algorithms and further certain conditions.

Section 3.1.3 gives the resulting module sequence the right characteristic of an ATE element profile by setting the proportions of each module in the element structure. Then we map the predefined module profile (see definition 2.3.2) to corresponding vector in the element structure, and finally discuss on complex ATEs structure which could not be reconstructed using the principle construction rule as well as posting solutions for the reconstruction of such complex elements.

3.1.1 General Algorithm

In chapter 2 we have seen some introduction of the properties of ATE module sequence, and the way each components or modules are combined together using the principle rule for reconstructing the sequence. We also know that the rule returns a correct sequence structure when reconstructing a certain type of element with simple composition under basic conditions. In this section we are going to describe the principle rule as well as the conditions necessary for using the principle rule for reconstruction ATE module sequence more implementation-oriented and mathematically respectively. Let us firstly have a closer look at the principle rule presented in section 2.3.2.

The directed graph $G = (V, E)$ is designed to control the combinatorial process for reconstruction of the ATE module sequence $S_{ATE} = (t_1, ..., t_i, ..., t_N)$ where $t = \{a, b, c, d, e, f\}$ (see definition 2.3.4). In order to facilitate the implementation of this construction rule in our algorithm, we associate graph *G* with its adjacency matrix,

denoted by $\varphi(v_{jk})$ which consists of 6×6 elements $v_{jk}=\{0,1\}$ with $1\le j,k\le 6$, and is presented as shown below.

$$\varphi(v_{jk})=\begin{pmatrix} v_{11} & \cdots & v_{16} \\ \vdots & \ddots & \vdots \\ v_{61} & \cdots & v_{66} \end{pmatrix}= \begin{array}{c} \\ a^j \\ b^j \\ c^j \\ d^j \\ e^j \\ f^j \end{array} \begin{array}{c} \begin{array}{cccccc} a^k & b^k & c^k & d^k & e^k & f^k \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{3.1}$$

In relation to the definition of digraph G presented in section 2.3.2 and its adjacency matrix $\varphi(v_{jk})$ above, we see that $\sum\varepsilon=\sum v_{jk}=10$ where edges $\varepsilon\in G$ and $v_{jk}\in\varphi$ with $v_{jk}=1\,|\,\forall j,k=1,...,6$ representing the allowed link between the end point e^{t_i} of module $t_i\in S_{ATE}$ and the start point $s^{t_{i+1}}$ of the adjacent module $t_{i+1}\in S_{ATE}\,|\,\forall i=1,...,N$. The reconstruction process begins with adding the first selected module t_1 to the sequence S_{ATE}. Its end point e^{t_1} is then referred to the row vector t_1^j of the corresponding type *t* in order to check for the next allowed module type t_2 by reading the values of each elements $v_{jk}\in t_1^j\,|\,k=1,...,6$, and only add that next module type t^k to S_{ATE} if $v_{jk}=1$. The column index *k* at the reconstruction step $i=2$ represents the start point s^{t_2} to be linked to e^{t_1} as well as representing the end point e^{t_2} at the same time. Now the process can begin anew by referring the e^{t_2} to the row vector t_2^j, checking the value of $v_{jk}\in t_2^j\,|\,k=1,...,6$, and add t_3 to the sequence if $v_{jk}=1$, and so forth until $i=N$. Before we describe the reconstruction procedure in detail we need to focus on some conditions which are required for the implementation of this construction rule.

For the reason that all modules, except the connector *c*-module, are very much restricted in connection with others which can be seen from the values $v_{jk} = 0$ in φ, it is necessary that the input data, in this case input modules for reconstruction of S_{ATE}, are defined correctly. Recalling the architectural function of the modules discussed in 2.2.3 and the definition of the module sequence of ATE in section 2.3.2 we know that a typical ATE structure or sequence $S_{ATE} = (t_1,...,t_i,...,t_N)$ consist of modules of type $t = \{a,b,c,d,e,f\}$ represented by a sequence of a finite set of vertices $v_i \in W(G)$ $\forall i = 1,...,N$ where *W* denotes the directed walk of the graph *G* and *N* the total number of *t* in S_{ATE}. The component of type t can occur multiple at once in the sequence S_{ATE} and is defined by $n_t t$ where n_t denotes the coefficient of the component *t*. Thus the sum *N* of all components $t \in S_{ATE}$ is calculated as: $N = n_a a + n_b b + n_c c + n_d d + n_e e + n_f f = \{n_t t\}$ for all $t = a,b,c,d,e,f$ and $n_t \in \mathbb{N}_{\geq 0}$. Additionally, we define a subsequence $S' \subset S$ with subset $N' \subset N$. Based on analysis of the ATE architectural vocabulary in section 2.2.3, we can define the general condition for reconstruction of ATE structure as following:

Definition 3.1.1 (general conditions)

ATE sequence S_{ATE} can only be properly reconstructed with input components that satisfy the following general conditions:

1. *If $n_b + n_f \neq 0$, then there must exist the corresponding a-module to be linked to b- and f-module:* $3 \geq n_a \geq n_b + n_f$.
2. *Modules of type* b,e and f *typically occur single in* S_{ATE}*:* $n_b, n_e, n_f \leq 1$.
3. *The number of the connector c-module depends on the number of the convex modules* a,d and e *to be connect with, and is calculated by:* $n_c = n_a + n_d + n_e - 1$.

4. *The sequence* S_{ATE} *generally begins and terminates with an a-module:* $t_1 = a_1$ and $t_N = a_{n_a}$, *we obtain* $S_{ATE} = (a_1, S', a_{n_a})$.
5. *d-module can be selected freely and independent from others:* $n_d \geq 0$

The first condition makes sure that with the existence of *b* and *f*, the block (*a, b*) and (*f, a*) can be formed. Suppose a sequence *S* consists of input modules: $N = \{a, b, n_c c, 2d, e, f\}$ where $n_a = 1 < n_b + n_f$ is to be constructed. Base on the 3rd condition we obtain: $n_c = n_a + n_d + n_e = 1 + 2 + 1 = 4$, thus $N = \{a, b, 4c, 2d, e, f\}$ with a total number of input modules $N = 10$. The sequence begins with forming the (*ab*) block (see 4th condition), and incorrectly terminate with an *f*-module: $S = (a, b, c, d, c, d, c, e, c, f) \neq S_{ATE}$. Because $\{S = (a, b, S', f), S = (b, S', f, a), S = (b, S', f)\} \notin S_{ATE}$, there must exist a sufficient number of *a*-modules in relationship with the availability of *b*- and *f*-module. We have pointed out in section 2.2.3 that the complexity of ATE structure depends on the number *d*-module, thus the coefficient n_d, where the others are restricted by a certain limitation. The Second condition ensures that the number of *b*-, *e*- and *f*-module in S_{ATE} is limited to 1 of each type.

The third condition allows us to calculate the required number n_c of the connector *c*-module to provide sufficient links between all given convex module types *a, d* and *e*. Although *b* and *f* are also convex modules, they do not require additional connector because as *b* and *f* exist as input module, they are supposed be directly connected with the corresponding a_1 and a_2 respectively (1st condition). This means that c_1 and c_2 which were reserved for a_1 and a_2 are now used for connecting (a_1, b) and (f, a_2) to $S' \subset S_{ATE}$. The fourth condition ensures S_{ATE} to be initialized with $a = a_1$ and reserve one $a = a_N$ for completing or terminating the sequence.

Incorporating these five conditions with the adjacency matrix shown in (3.1) we proceed with setting up a *general algorithm* for reconstruction of Angkor Temple Elements.

Let the row vector $t_i^j \in \varphi$ denoting the end point e^{t_i} of the module type *t*, and the row index $j(t_i)$ of the matrix (2.5), that is currently positioned at the i^{th} index in the sequence to be reconstructed $S_{ATE} = (t_1,...,t_i,...,t_N)$ with $N \in \mathbb{N}^*$ input modules. Based on matrix $\varphi(v_{jk})$ we want to find out the following allowed module t_{i+1}, of which its start point $s^{t_{i+1}}$ to be linked with e^{t_i} is presented by the column vector t_{i+1}^k at the index $k(t_{i+1})$, and add it to the sequence. We define S_{Temp} denoting a temporal sequence while the reconstruction is still in process, thus while $i < N$.

The general reconstruction algorithm consists of four steps:

Algorithm 3.1.1 (general algorithm: LPE reconstruction)

For all components (input modules) $t_i \in S_{ATE} \quad \forall i = 1,...,N$ do the following:

1. As the forth condition implies, initialize S_{Temp} with $t_1 = {}_1a_1$ at $i = 1$. We refer the row vector $t_i^j(\varphi) = a^j$ at the row index $j(t_i) = j({}_1a) = 1$. Else quit the process.
2. For all elements $v_{jk} = \{0,1 | \forall k = 1,...,6\} \in t_i^j(\varphi)$, do the following:
    a. Add the module $t_{i+1} \equiv t^k$ to S_{Temp} of which the type is defined by the column index $k(t_i)$ if $v_{jk} = 1$.
    b. Refer the row index of $j(t_{i+1}) = k(t_i)$.
3. Loop step 2 for $i+1$ iteration until $i = N$, such that $k(t_{i+1}) = k(t_i)+1,...,6$ and loop if $v_{jk} = 0 | \forall k(t_{i+1}) = k(t_i)+1,...,6$.
4. Set $S_{ATE} = S_{Temp} = (t_1,...,t_i,...,t_N)$ if $i = N$.

Algorithm 3.1.1 comprises of two considerable properties in relationship with the ATE architecture. Firstly, is the looping process of generating the (*c,d*) block until $n_d = 0$, and secondly ${}_2a$ is reserved to add at the end of the sequence, thus $t_N = {}_2a$. We would like to discuss these phenomena with concrete examples on reconstruction of ATE using the general algorithm.

For the analysis of the reconstruction behavior especially the priorities of module types for selection while implementing the general algorithm, we suppose to have an unlimited resource of module of any type as input. In this case, the first, second and the third points of the general condition is temporary not necessary but we take into account of the fourth. Figure 3.1.1 demonstrates the reconstruction procedure.

It is important to notice that after adding the satisfied component t_{i+1} at the *k*-th column index of t_i^j in the reconstruction phase (a) of step 2, we do not reset $k = 1$. This means that the system continues evaluating the value v_{jk} of t_{i+1}^j from position $k(t_i)+1$. At the end of the first iteration of *k*-loop, thus at the step $j_i = 3$ for evaluating components to be linked to module c_3, t_{i+3} would be a_2 if the system evaluated $v_{jk} \in t_{i+3}$ anew from $k = 1$ because $v_{ca} = v_{31} = 1$. Instead $t_{i+3} = d$ because $k(t_{i+2}) = 3$ thus the system continues evaluating $k(t_{i+3}) > 3$, and select *d* since $v_{jk} = v_{cd} = v_{34} = 1$ being the next approved module. In addition, base on the values $v_{j1} \in a^k \,|\, j = 1,...,6$ we see that only *c*- and *f*-module are allowed to link to *a*. This means that ${}_2a$ is not focused until *f* is added to S_{Temp}. This is one of the fact that prevents from adding ${}_2a$ in between the sequence, however let us discuss how the (*c,d*) block is generated.

During the step of $t_i = d_4$, the system must loop to the second $k = 1,2,3,4$ iteration because $v_{dk} = 0 \,|\, \forall k = 5,6$, and accept ${}_2c$ at $k(t_4) = 3$, then repeating the same procedure as shown in step $i = 3$ forming repeatedly a sequence of (*c,d*) blocks n_d times before switching to *e* then *f*, and loop to *a* again.

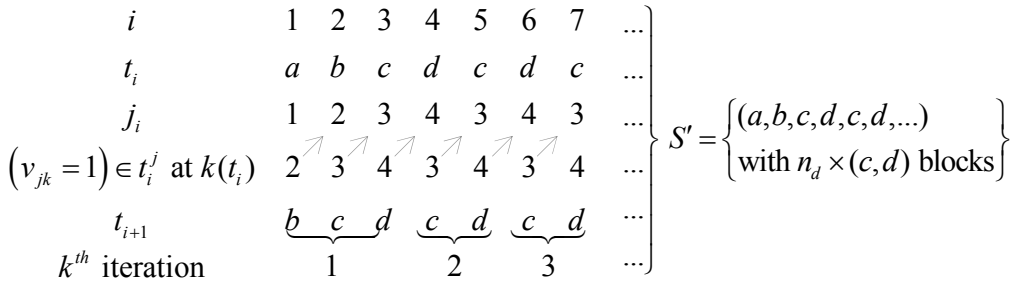


Figure 3.1.1
The reconstruction procedure using the general algorithm for reconstructing a sequence a subsequence S'

For this reason the (*c,d*) blocks is automatically generated, and ${}_2a$ remains unused until $i = N-1$. We consider in figure 3.1.1 that $S' = (t_1, t_2, ..., t_{(2n_d+3)})$ $= \left[a_1, b_2, (cd \text{ blocks}), c_{(2n_d+3)} \right]$ is already a correct sub-sequence of an ATE, which correspond to the sequence of a column-capital or wall-capital. Our hypothesis for this analysis was an unlimited supply of resources of all module types, thus we would obtain a sub-sequence S' with infinite set of (*c,d*) blocks. On the other hand, the row vectors $e^j = d^j$ and $e^k = d^k$ with $\left\{ d^j, e^j, d^k, e^k \right\} \in \varphi$, thus similar to the case of forming of (*c,d*) blocks, the algorithm would reconstruct unwanted (*c,e*) blocks. For this reason we now provide a limited number of resources for reconstruction.

Recalling the general condition of ATE sequence, we know that $3 \geq n_a \geq n_b + n_f$ and $n_b, n_e, n_f \leq 1$. From these three types we will provide a maximum, and three from the "repeated module" *d* thus $n_d = 3$. We obtain the input components for reconstruction of the sequence as following: $N = \{3a, b, n_c c, 3d, e, f\}$. We firstly need to calculate the required $n_c = n_a + n_e + n_d - 1 \ = 3+3+1-1 = 6$, thus $N = \{3a, b, 6c, 3d, e, f\} = 15$. Figure 3.1.2 shows that the general algorithm provides a sequence as following:

$$S = ({}_1a, {}_1b, {}_1c, {}_1d, {}_2c, {}_2d, {}_3c, {}_3d, {}_4c, {}_1e, {}_5c, {}_1f, {}_3a, {}_6c, {}_3a)$$

which can be considered as a 80% correct ATE sequence. The error lays in the mismatch between the temple element type to be reconstructed and the corresponding composition which we did not take into account in advance for the example above. An ATE sequence with both, b and f as well as $n_a = 3$ is normally a basement that requires specific additional treatments to the general condition which we will present later in this section.

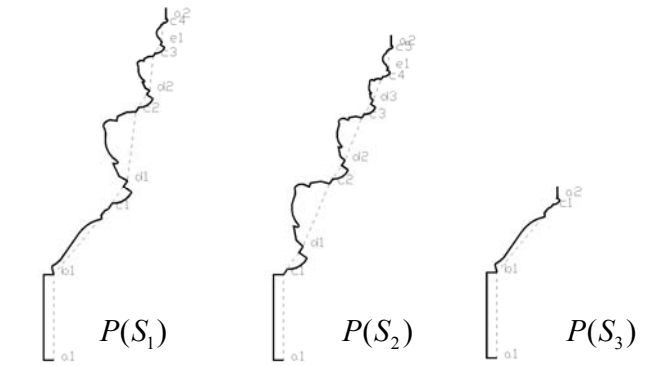
i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
t_i	${}_1a_1$	${}_1b_2$	${}_1c_3$	${}_1d_4$	${}_2c_5$	${}_2d_6$	${}_3c_7$	${}_3d_8$	${}_4c_9$	${}_1e_{10}$	${}_5c_{11}$	${}_1f_{12}$	${}_2a_{13}$	${}_6c_{14}$	${}_3a_{15}$
j_i	1	2	3	4	3	4	3	4	3	5	3	6	1	3	1
$\left(v_{jk}=1\right)\in t_i^j$ at $k(t_i)$	2	3	4	3	4	3	4	3	5	3	6	1	3	1	0
t_{i+1}	b_2	c_3	d_4	c_5	d_6	c_7	d_8	c_9	e_{10}	c_{11}	f_{12}	a_{13}	c_{14}	a_{15}	0
k^{th} iteration	1			2		3		4		5		6		7	

Figure 3.1.2
A procedure for reconstruction of a sequence *S* using $N = \{3a, b, 6c, 3d, e, f\} = 15$ input modules, and resulting $S = ({}_1a, {}_1b, {}_1c, {}_1d, {}_2c, {}_2d, {}_3c, {}_3d, {}_4c, {}_1e, {}_5c, {}_1f, {}_3a, {}_6c, {}_3a)$.

Let us restructure *S* to become an ATE module sequence. To do this, we need to dispose superfluous modules including $f = {}_1f$, $a = {}_3a$ and the accompanying $c = {}_6c$. We then obtain a correct ATE sequence that corresponds to a column- or wall-base element:` $S_{ATE} = ({}_1a, {}_1b, {}_1c, {}_1d, {}_2c, {}_2d, {}_3c, {}_3d, {}_4c, {}_1e, {}_5c, {}_2a)$ where again the (*c,d*) blocks are automatically generated, and ${}_2a$ is placed at the end of the sequence, as we implied earlier in this section.

Using the general algorithm we are able to properly reconstruct sequences of ATE, not only the column-base and wall-base elements but also any other elements in the group of the lower part elements (see section 2.2.2). This algorithm is also applicable for reconstruction of elements with following properties: $i({}_{n_b}b) < i({}_{n_d}d) < i({}_{n_e}e) < i({}_{n_f}f)$. Figure 3.1.3 presents additional samples of ATE types of which reconstruction are based on general algorithm. However, we are still not able to apply the general algorithm solely

for reconstructing temple elements such as the basement, column- or wall-capital, colonnette, window-grids, etc. The structure of these element types are either *non-alphabetic ordered sequence* or *special complex elements*. Properties of special elements will be discussed in detail in section 3.1.6. For now let us focus on elements with non-alphabetic sequence.



S_1

$N(S_1) = \{2a, b, 4c, 2d, e\}$

S_2

$N(S_2) = \{2a, 5c, 3d, e\}$

S_3

$N(S_3) = \{2a, b, c\}$

Figure 3.1.3
Samples for S_{ATE} reconstruction using the general algorithm, and their corresponding temple element profiles $P(S_{ATE})$ where $P(S_1)$ and $P(S_2)$ are samples of a typical wall-base or column-capital. $P(S_3)$ is a very simple structure of a wall-base of Angkor Wat Tower tier.

There are several sorts of ATE with non-alphabetic sequence found on Angkor temple including the basement, wall-capital, column-capital, cornice of the tower tier, etc. But as we implied in section 2.2.3, most of them corresponds to only two groups, the

group of the UPE and basement. Above we have seen an algorithm for reconstructing ATE in the group of LPE, and do not correspond to the non-alphabetic sequence. Reconstructing the sequence of a non-alphabetic element is based on the general algorithm but with additional modifications. The following we present properties of non-alphabetic elements and the required modifications to the general algorithm by examine each element by their type individually.

3.1.2 Non-alphabetic Sequence of ATE

A special feature of elements with non-alphabetic module sequence is the uncommon position of the *e*-module and ${}_3a$ in its structure. The example for reconstructing S_{ATE} using the general algorithm in figure 3.1.3 shows that *e* is positioned at the index $i(e) > i({}_{n_d}d)$, and ${}_3a$ at $i({}_3a) = N$. On elements with non-alphabetic sequence however, they are positioning in a different order.

Definition 3.1.1 non-alphabetic sequence

Angkor temple elements of which modules including ${}_{n_b}b, {}_{n_d}d, {}_{n_e}e, {}_{n_f}f$ and ${}_{n_a}a$ *occur in its sequence* S_{NAS} *not in an alphabetic order counting from left to right (on element profile: upwards), thus the index* $\left\{i({}_{n_b}b) < i({}_{n_d}d) < i({}_{n_e}e) < i({}_{n_f}f) < i({}_{n_a}a)\right\} \notin S_{NAS}$ *are called non-alphabetic sequence.*

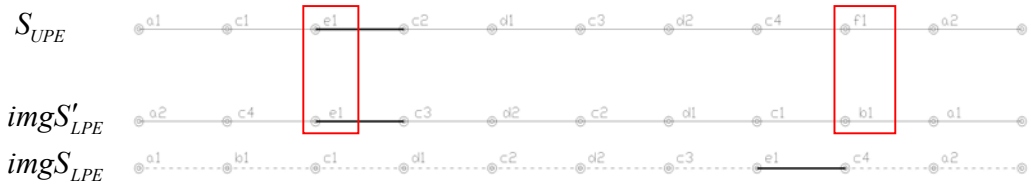
In the structure of an UPE column-capital or wall-capital we can observe that $i({}_{n_e}e) < \left[i({}_1d) < i({}_{n_d}d)\right] < i({}_{n_f}f) < i({}_{n_a}a)$, and the structure of a basement typically $i({}_{n_b}b) < i({}_1d) < \left[\begin{matrix} i({}_{n_e}e) \\ i({}_{n_a}a) \end{matrix}\middle| = \frac{N+1}{2}\right] < i({}_{n_d}d) < i({}_{n_f}f)$ with $N = 2x \middle| \forall x \in \mathbb{N}^{\geq 0}$ where the *e*-module occur directly before the (*c,d*) block in a UPE structure, and *e*-module or ${}_{n_a}a = {}_3a$ in the middle of the whole sequence for the case of a basement. Thus the

modification has to be carried out differently, forming two different algorithms. In the next step we present the algorithm for reconstruction of upper part elements (UPE), and will begin with analyzing their properties.

Upper Part Element (UPE)

Theoretically, the module sequence S_{UPE} of an UPE is equivalent to the *invert sequence* $imgS'_{LPE}$ of an imaginary sequence $imgS_{LPE}$ of a corresponding LPE (see figure 2.2.9 in section 2.2.3). Together $S_{UPE} \bigcap imgS_{LPE} = imgS_{basement}$ they form up a sequence of an imaginary basement with the following conditions:

$$\left.\begin{array}{l} n_t = n_{img(t)} \,\big|\, \forall t = \{a,b,d,e,f\} \text{ with } t \in S_{UPE} \text{ and } img(t) \in imgS_{LPE} \\ f(S_{UPE}) \equiv b(imgS_{LPE}) \\ e(S_{UPE}) \mapsto e(imgS_{LPE}) \text{ or } {}_1a(S_{UPE}) \mapsto {}_2a(imgS_{LPE}) \end{array}\right| \Rightarrow imgS_{basment}$$



$$S_{UPE} = imgS'_{LPE} = (a_1, c_1, e_1, c_2, d_1, c_3, d_2, c_4, f_1, a_2) \text{ with } f_1 \equiv b_1$$

Figure 3.1.4
A sample for the equivalent sequences $S_{UPE} = imgS'_{LPE}$ with $f_1 \equiv b_1$ and $N(S_{UPE}) = \{2a, 4c, 2d, e, f\}$ and $N = 10$.

Base on this theory and the inverse-directed priority of modules in S_{LPE} we can define that $i({}_{n_e}e) < i({}_{n_d}d) < i({}_{n_f}f)$ where *e*-module has priority over *d*-module. On the other hand, in the matrix $\varphi(v_{jk})$ (3.1) we see that the row vector $d^j = e^j$ and column vector $d^k = e^k$. These properties allow us to simply change the order of d^j with e^j and

d^k with e^k in order to give priority to *e*-module over *d*-module in the for loop procedure. We obtain a new matrix $\varphi_{UPE}(v_{jk}) = \varphi(v_{jk})$ of which the row- and column vectors are referred to $t_i^j = t_i^k = \{a_1, b_2, c_3, e_4, d_5, f_6\} \big| \forall i = 1,...,6$ (see matrix (3.2)).

The algorithm for reconstructing S_{UPE} is based on the general condition and has the same procedure as that of the general algorithm but with following additional condition:

- $n_a \le 2,\ n_b = 0$
- In step 2 of the general algorithm we replace $\varphi(v_{jk})$ with $\varphi_{UPE}(v_{jk})$

$$\varphi_{UPE}(v_{jk}) = \begin{pmatrix} v_{11} & \cdots & v_{16} \\ \vdots & \ddots & \vdots \\ v_{61} & \cdots & v_{66} \end{pmatrix} = \begin{array}{c} \\ a^j \\ b^j \\ c^j \\ e^j \\ d^j \\ f^j \end{array} \begin{array}{c} \begin{array}{cccccc} a^k & b^k & c^k & e^k & d^k & f^k \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{3.2}$$

We obtain a new algorithm:

Algorithm 3.1.2 (UPE reconstruction)

For all components (input modules) $t_i \in S_{UPE} \quad \forall i = 1,...,N$ do the following:

1. As the forth condition implies, initialize S_{Temp} with $t_1 = {}_1a_1$ at $i = 1$. We refer the row vector $t_i^j(\varphi_{UPE}) = a^j$ at the row index $j(t_i) = j({}_1a) = 1$. Else quit the process.
2. For all elements $v_{jk} = \{0,1 \big| \forall k = 1,...,6\} \in t_i^j(\varphi_{UPE})$, do the following:

a. Add the module $t_{i+1} \equiv t^k$ to S_{Temp} of which the type is defined by the column index $k(t_i)$ if $v_{jk} = 1$.

b. Refer the row index of $j(t_{i+1}) = k(t_i)$.

3. Loop step 2 for $i+1$ iteration until $i = N$, such that $k(t_{i+1}) = k(t_i) + 1,...,6$ and loop if $v_{jk} = 0 \big| \forall k(t_{i+1}) = k(t_i) + 1,...,6$.

4. Set $S_{UPE} = S_{Temp} = (t_1,...,t_i,...,t_N)$ if $i = N$.

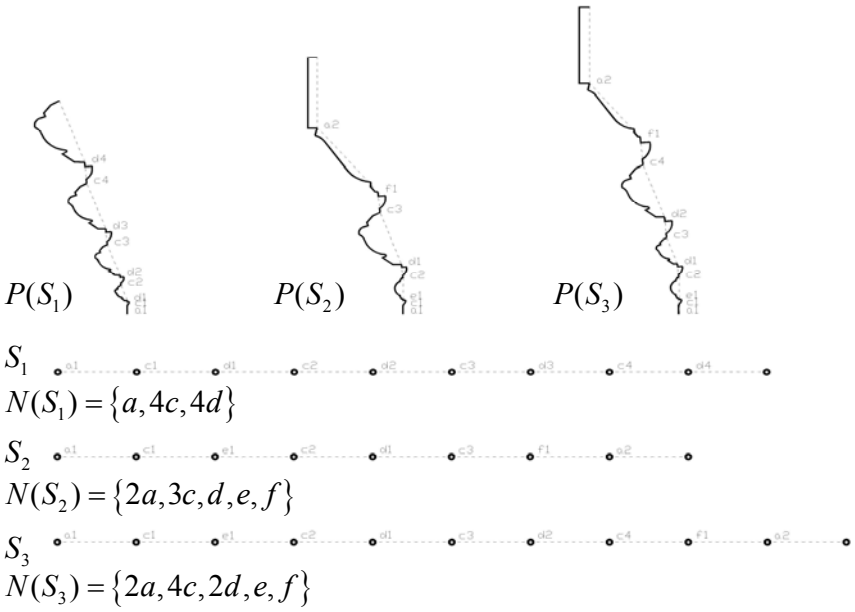


Figure 3.1.5
Three samples of reconstruction of upper part elements sequence including wall-capital and cornice using algorithm 3.1.2, and their corresponding profiles (the three images in the top). $P(S_1)$ is a profile of Angkor Wat wall-capital, $P(S_2)$ is a wall-capital Chao Say Tevda temple and $P(S_3)$ a cornice of the first stage of the Thomannon temple tower.

Basement

The basement element is the most complex "single element", and covers up to 50% of the whole temple complex surface in particular on Temple Mountain such as Angkor Wat, Bapuon temple, Takeo temple, etc. It is the only element with the highest number of *a*-

module: $n_a = 3$ where two of them, as common on ATE, locating at both ends of its module sequence structure, and the potential third one in the middle. The architecture of a basement is in terms of module sequence vertically symmetric, again see figure 2.2.9. This property is responsible for the architectural relationship between basement and LPE/UPE. Since one module must be locating in the middle of the sequence, the number of its components is constantly odd. As consequence, we can observe that *b*- and *f*-module must exist simultaneously in a basement module sequence.

Another considerable property of a basement is that it is the only one temple element which are (almost) consisting of two different (*cd*) blocks (if $n_d = 2x \left| \forall x \in \mathbb{N}^{\geq 0} \right.$): one in the lower part and another in upper part, separated by either *a*, *c*, *d* or *e*-module, exactly in middle of the its sequence which are called the *central module* (see figure 3.1.6).

Definition 3.1.2 (central module)

The number of components $N(S_{basement})$ *of a basement sequence* $S_{basement} = (t_1,...,t_i,...,t_N)$ *is always an odd number* $N = 2X + 1 \left| \forall X \in \mathbb{N}^{>0} \right.$. *The module* t_i *locating at the index* $i = \frac{N+1}{2}$ *in* $S_{basement}$ *is called the central module.*

In order to describe the structure of a basement module sequence $S_{basement}$ with *N* components, again let us define $S' \subset S_{basement}$ to be a sub-sequence of $S_{basement}$. There are several possible structures of Angkor temple basement, but they can be generally described as below:

$$S_{basement} = \left(S'_1, \left[\frac{n_d}{2}(cd)blocks \right], S'_2, \left[\frac{n_d}{2}(cd)blocks \right], S'_3 \right)$$

with $S'_1 := \{a,b\}$, $S'_2 := \left\{ (c, a_{\frac{N+1}{2}}), (c, d_{\frac{N+1}{2}}), (c, e_{\frac{N+1}{2}}), \varnothing \right\}$ and $S'_3 := \{a,c,f\}$

and consists of the following properties:

1. $n_a \le 3$ and $1 \ne n_b + n_f \le 2$
2. If $n_d = 2X_d + 1$ and $n_a = 2$, $n_e = 0$ then $S_2' = (c,d)$, else if $n_a + n_e = 2X + 1$ and $n_d = 2X_d$ then $S_2' = (c,t)$ with $t = \{a,e\}$, else if $n_t = 2X$ with $t = \{a,d,e\}$ then $S_2' = \varnothing$, and the $c_{\frac{N+1}{2}}$ becomes the central module for all $X, X_d \in \mathbb{Z}^+$.

Figure 3.1.6
Two types basement of Chao Say Tevda temple, left: $S_2' = (c,a)$, right: $S_2' = \varnothing$.

Lemma 3.1.3. The input modules $N(S_{basement}) = 2X + 1 \,|\, \forall X \in \mathbb{N}^{>0}$ for reconstruction of Angkor temple basement sequence $S_{basement}$ must be selected base on following pattern:

$$\begin{aligned} &1.\quad n_a = 2, \begin{cases} a). \rightarrow n_e = 0, & \begin{cases} i). \rightarrow n_d = 2X_d + 1 \\ ii). \rightarrow n_d = 2X_d \end{cases} \\ b). \rightarrow n_e = 1, & n_d = 2X_d \end{cases} \\ &2.\quad n_a = 3, \qquad n_e = 0, \qquad n_d = 2X_d \end{aligned} \quad \Big| \quad \forall X_d \in \mathbb{N}^{\ge 0} \qquad (3.3)$$

Proof: The input module *e* is depending on *a*, and *d* depends on *e* respectively. We want to proof that for any case in (3.3), $N(S_{basement})$ remains being an odd number. We consider in the first property of a basement that $n_b + n_f = \{0,2\} = even$ thus $N(S_{basement})$ does not depend these two module types. We study only the case of the rest. The first

case of $n_a = 2$ there exists two cases of *e*: if n_e is even, n_d is set to odd, else the other way round. We obtain:

$$1)a)i): \quad \begin{vmatrix} n_c = n_a + n_e + n_d - 1 = \left[2 + 0 + (2X_d + 1)\right] - 1 = 2(X_d + 1) = 2X_c \\ \Rightarrow N(S_{basement}) = n_a + n_c + n_e + n_d = 2 + 2X_c + 0 + (2X_d + 1) \\ = 2(2X + 1) + 1 = odd \end{vmatrix}$$

$$1)a)ii): \quad \begin{vmatrix} n_c = n_a + n_e + n_d - 1 = \left[2 + 0 + 2X_d\right] - 1 = 2(X_d + 1) - 1 = 2X_c - 1 \\ \Rightarrow N(S_{basement}) = n_a + n_c + n_e + n_d = 2 + (2X_c - 1) + 0 + 2X_d \\ = 2(2X) + 1 = odd \end{vmatrix}$$

$$1)b): \quad \begin{vmatrix} n_c = n_a + n_e + n_d - 1 = \left[2 + 1 + 2X_d\right] - 1 = 2(2X_d + 1) = 2X_c \\ \Rightarrow N(S_{basement}) = n_a + n_c + n_e + n_d = 2 + 2X_c + 1 + 2X_d \\ = 2(2X + 1) + 1 = odd \end{vmatrix}$$

In the fourth case of $n_a = 3$, the module *e* do not exist thus $n_e = 0$, and n_d must be even in order to form the lower- and the upper part (*cd*) blocks. In this case we have *a* as the central module and we obtain:

$$2. \quad \begin{vmatrix} n_c = n_a + n_e + n_d = \left[3 + 0 + 2X_d\right] - 1 = 2(2X_d + 1) + 1 - 1 = 2X_c \\ \Rightarrow N(S_{basement}) = n_a + n_c + n_e + n_d = 3 + 0 + 2X_c + 2X_d \\ = 2(2X + 1) + 1 = odd \end{vmatrix}$$

The characteristic of the basement is there for depending on the set of input modules shown in (3.3). Consequently, the algorithm for reconstruction of Angkor temple basement element must evaluate input data in order to define the case, respectively the type of the basement structure, to be reconstructed in advance.

It is for this reason not possible to solely implement the general algorithm which would produce one single (*cd*) block for $n_d = 2X \left| \forall X \in \mathbb{N}^{>0} \right.$ .Before we present the algorithm for reconstructing a basement, we would like to recall that a basement structure consists of two (*cd*) blocks which are separated by a central module. However, we consider that the case in 1)*a*)*ii*) there is no convex module amid the (*cd*) block. In this case *c*-module becomes the central module of the sequence, see the right hand side image

of figure 3.1.6. Similarly case can be seen in 1)*a*)*i*). Although the structure here comprising of a convex module as its central module, which is the *d*-module, its (*cd*) block remains unified as one single block. In both cases the priority of modules to be checked in the reconstruction process (range in the matrix) remains as unchanged. The uppermost basement of the central east-west gallery of Angkor Wat cruciform is a good example for this case. Thus we could simply use the general algorithm to reconstruct their sequence without any modification to the procedure or the matrix. But basement of these two types can be rather seldom found on Angkor temples, and if exist then mostly as small scale basements such as shown as examples above.

More complicated are the other two cases: 1)*b*) and 2, where the (*cd*) block are separated by a convex module: *a*- and *e*-module. Our proposed solution is to give priority to these central modules ($_2a$ and $_1e$) over $_{\frac{n_d}{2}+1}d_{\frac{N+1}{2}}$, the first *d*-module of the upper part (*cd*) block with identity $\frac{n_d}{2}+1$ and index $\frac{N+1}{2}$ where n_d and N respectively denoting the number of *d*-modules and the total number of components in the basement.

The algorithm for reconstructing the basement is directed to the actual selective case from the pattern (3.3), and can be described as following 5 steps:

Algorithm 3.1.3 (basement reconstruction)

For all components (input modules) $t_i \in S_{basement} \quad \forall i = 1,...,N$ do the following:

1. Select case in following pattern:

$$\left.\begin{array}{lll} 1. & n_a = 2, & \begin{cases} a). \rightarrow n_e = 0, & \begin{cases} i). \rightarrow n_d = 2X_d + 1 \\ ii). \rightarrow n_d = 2X_d \end{cases} \\ b). \rightarrow n_e = 1, & n_d = 2X_d \end{cases} \\ 2. & n_a = 3, & n_e = 0, \qquad n_d = 2X_d \end{array}\right| \quad \forall X_d \in \mathbb{N}^{\geq 0}$$

2. Initialize S_{Temp} with $t_1 = {}_1a_1$ at $i = 1$, and refer the row vector $t_i^j(\varphi) = a^j$ at the row index $j(t_i) = j({}_1a) = 1$ for $n_a \geq 2$, else quit the process.
3. For all elements $v_{jk} = \{0,1 | \forall k = 1,...,6\} \in t_i^j(\varphi)$, do the following:

a. If step 1 be case 1)*b*) then set $t_{\frac{N+1}{2}} = e$, add $t_{\frac{N+1}{2}}$ to S_{Temp} and skip c).

b. If step 1 be the case 2 then set $t_{\frac{N+1}{2}} = {}_2a$, add $t_{\frac{N+1}{2}}$ to S_{Temp} and skip c).

c. Add $t_{i+1} \equiv t^k$ to S_{Temp} of which the type is defined by the column index $k(t_i)$ if $v_{jk} = 1$.

d. Refer the row index of $j(t_{i+1}) = k(t_i)$.

4. Loop step 3 for $i+1$ iteration until $i = N$, such that $k(t_{i+1}) = k(t_i) + 1,...,6$ and loop if $v_{jk} = 0 \left| \forall k(t_{i+1}) = k(t_i) + 1,...,6 \right.$.

5. Set $S_{basement} = S_{Temp} = (t_1,...,t_i,...,t_N)$ if $i = N$.

We see that there is no any change to the matrix as the case of UPE. The first step does not only make sure that the composition is properly defined but also describes the varying cases which control further step in the algorithm. As usual, step 2 initializes the sequence with an initiator *a*. Special feature in step 3 is the case selection defined in a). and b). where either *a* or *e* is set as the central module of the sequence.

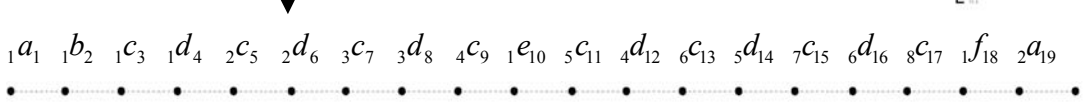
As soon as the lower part sequence of the basement is added to S_{Temp} and the $t_i = t_{\frac{N+1}{2}}$ is evaluating, the algorithm automatically apply the central module $t_i = t_{\frac{N+1}{2}} = \left| \begin{matrix} a \\ e \end{matrix} \right.$ depending on the actual case. For both cases the system does not need to evaluate the values of φ at $j_{\frac{N-1}{2}}$, thus skipping phase c) of step 3. Then the reconstruction procedure can follow up its ordinary process until the upper part sequence is completed. Figure 3.1.7 gives an example on implementing the Algorithm 3.1.3.

Figure 3.1.7

Reconstruction of a basement composing of $N(S_{basement}) = \{2a, b, n_c c, 6d, e, f\}$. We first calculate $n_c = n_a + n_d + n_e - 1 = 2 + 6 + 1 - 1 = 8$, and obtain an input $\Rightarrow N(S_{basement}) = \{2a, b, 8c, 6d, e, f\}$ with $N = 19$. The composition corresponds to case a) of step 3 where $n_e = 1$. Thus $t_{\frac{19+1}{2}} = t_{10} = {}_1e_{10}$. The reconstruction algorithm provides a sequence:

$$S_{basement} = (a_1, b_2, c_3, d_4, c_5, d_6, c_7, d_8, c_9, e_{10}, c_{11}, d_{12}, c_{13}, d_{14}, c_{15}, d_{16}, c_{17}, f_{18}, a_{19})$$

Sequence of $S_{basement}$ ▼ $P(S_{basement})$: Profile of $S_{basement}$ ▶



Unlike previous algorithms which can be implemented for any LPE or UPE, this algorithm is specialized for reconstructing only a basement. These three algorithms cover the reconstruction of the sequence of most common Angkor- as well as Khmer temple elements. There are several elements of which definitions do not belong to the three groups mentioned above but its sequence property is identical to one of them. For instance the crown element of the tower is a single element with architectural function as the LPE, of which sequence properties are identical to each other as well.

It is possible in such cases to use the algorithm for reconstructing LPE to reconstruct the sequence structure of a crown, see figure 3.1.8. Other cases could belong to the complex elements, an element that consist of more then one single elements such as the tower stage of Angkor Wat also shown in figure 3.1.8. In such case it is possible to reconstruct each corresponding single element structure separately using the algorithm for reconstructing UPE, and later combine them together. Other more special complex

elements are too complex or consisting of too many single elements that it make no sense to reconstruct them one by one and manually combine. We propose another solution for treating such elements, and will be presented later in section 3.2 and 3.3.

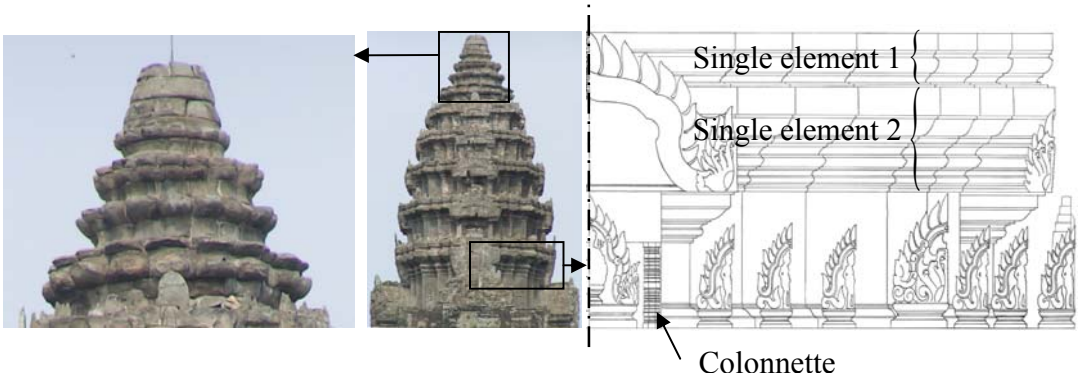


Figure 3.1.8
Angkor Wat central tower and its elements: left the crown, and the right image shows a drawing of the first tower stage, a special complex element consisting of 2 single UPEs (Drawing done by EFEO).

Until now we were able to reconstruct the sequences of modules in an Angkor temple elements structure. The modules sequences however, are represented by the vectors of each corresponding modules, and still appear as a sequence of horizontal homogeneous line segments. In the following step we are going provide advanced properties to the modules in order to form the structure into a proper state of an ATE.

3.1.3 Proportion of Modules in the Sequence

Our goal in this section is to generate a two dimensional profile of ATE in which each module vector appears with particular proportion of length relative to a unit length that we are going find out. In addition, as we have mentioned in section 2.2.2, also see section 2.3.1, that we need to give a certain angle value to the vectors in order to obtain an upright standing profile, and further responsible for the jut effect, see the mathematical model in figure 2.3.6. These two modifications, scaling and rotating the vector lines, will be discussed in the next section.

The proportion of the components in a module sequence of ATE is generally depending on the type of that element. Furthermore for each component in the sequence it differs from one to another simultaneously by both the scale l_t and angle value α_t. The procedure for modifying the module sequence (sequence of horizontal module vectors achieved in earlier section) is for this reason divided into two steps: firstly we set the components individually to the right size, and secondly we rotate each of them by the corresponding angle value. Similar to the reconstruction of the sequence, we analyze the proportions by type of element groups: the lower part elements (wall-base, column-base, etc.), the upper part element (wall-capital, column-capital, etc.) and the basement element. We begin this time with the basement, and study the scaling properties in advance before showing the properties for rotation. But first of all let us have a closer look at common characteristics of ATE in terms of proportion of their components.

We have mentioned in section 2.2.2 about the jut effect of ATEs at a certain part of the temple, and that it occurs at the transitional part between two different elements. Generally, we can observe that the closer a module of an element is located to that transitional part, the larger becomes its size. Another considerable characteristic is the size of *c*-module. From section 2.2.2 we also know that the "connector" *c* is responsible for providing a harmonic space between two different convex modules, and early in this chapter that the number of *c* modules depends on the number of a certain convex modules as well. Similarly to these properties, we discovered that the size of *c*-module in a module sequence of an ATE is proportional to its adjacent convex modules as well. For these reasons we are going to discuss the proportions of convex modules in advance.

Let us have a look at the repeated convex module *d.* We know that in a large scale ATE with comparative large number of modules, the *d*-modules respectively the (*c,d*) block is the key for the complexity. The size of each *d*-module in that (*c,d*) block obviously does not remain constant thus $l_{(i\,d)} \neq l_{(i+1\,d)} \,|\forall i = 1,...,n_d$. On the other hand, for the reason that $n_a \leq 3$ and $\{n_b, n_e, n_f\} \leq 1$, see definition 3.1.1. of general condition, we

need to examine the ratios of lengths of the modules a, b, e and f in relation to each other, whereas *d* and *c* are discussed individually for the case of a basement element.

Basement

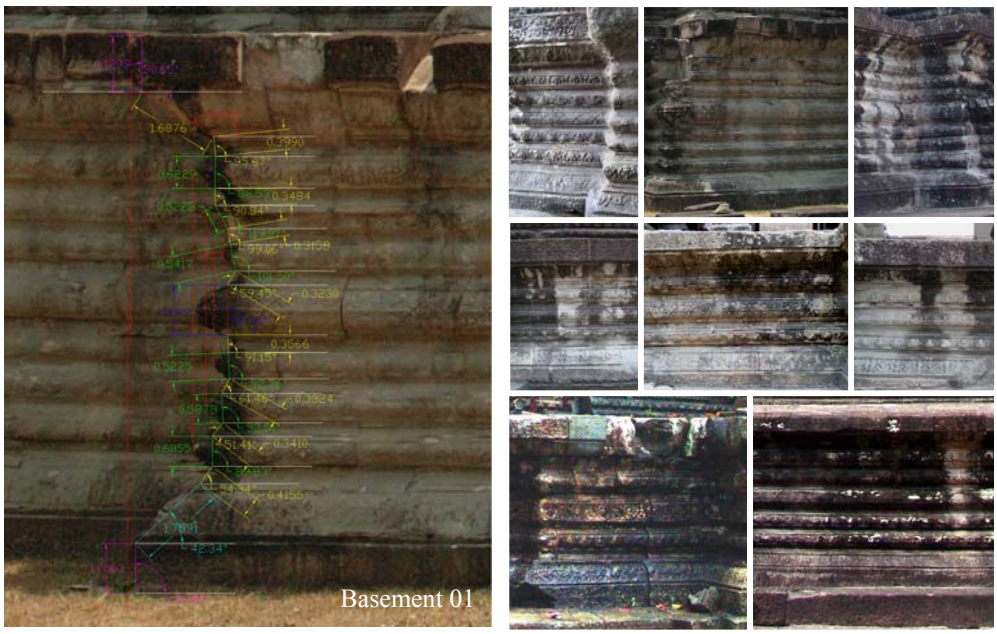
In order to examine the ratio of modules in an element we must set one module in that element sequence to be the unit to which ratios of all other modules are related. The selected one should exist in any type of temple element as well as in any case of one element type. Based on the general condition and the properties of LPE, UPE and the basement we have presented early in this chapter, we know that $\{n_b, n_e, n_f\} \leq 1$ and $n_d \geq 0$ which denoted that n_b, n_e, n_d and n_f can be zero. On the other hand the proportion of *c*-modules, as mentioned above, is not independent of other lengths, thus only the module of type *a* can be chosen as unit.

In addition, our investigation shows that special elements such as the wall capital of Angkor Wat shown image (a) of figure 3.1.9 comprises of only one *a* or $n_a = 1$ which is ${}_1a$. Generally the lower *a* module always exists but it partly vanishes in some cases because of the varying level of underneath lying basement. Image (b) and (c) give examples of such cases on basements of Chao Say Tevda and temple Angkor Wat respectively marked by red color hatching the vertical surfaces of ${}_1a$.

Figure 3.1.9
(a) Wall capital showing a single lower *a*-module (Angkor Wat), (b) Lower *a*-module partly vanishing shown by the thickness of the red colored hatching (Chao Say Tevda) and (c) Lower *a*-module vanishes or probably represented by the simple base below the basement.

Thus we set $_1a$ to be the base unit, and use it to study the proportions of other modules in the same module sequence of an ATE.

To do this we are going to capture the length l_t and angle α_t of module t from digital images of nine selected existing Angkor temple basements using AutoCAD, and compare their length- and angle function by groups. The length is given relative to the chosen base unit $_1a$.



Basement	Segment	*h*	*a*: l/α	*b*: l/α	*c*: l/α	*d*: l/α	*e*: l/α	*f*: l/α
01	Upper part	11.125			0.4/95.7			
					0.35/90.8	0.62/88.5		
					0.32/99.1	0.52/119.1		
			0.82/90.6		0.32/59.5	0.54/101.5		1.69/148
							1.01/90.4	
	Lower part				0.36/91.2			
					0.33/61.5	0.52/92.8		
					0.34/51.4	0.59/92.8		
			1/90	1.76/42.3	0.42/54.3	0.61/89.8		

Figure 3.1.10
Top images: Selected basements of Angkor Wat, Chao Say Tevda and Thomannon temple for capturing the lengths and angles (in degree) of their components. The measurements of the top left image is shown in table below it where the module length of $_1a$ is set as base unit.

Figure 3.1.10 shows the input data for our analysis and a demonstration of the measurements with the results in the table below. We used this method to measure all basements shown on the right side of this figure (see also appendix A.1). We must mention at this point that the manual measurement and the limited quality of our input image may cause minor errors. The parameters we obtained can be described in form of functions shown in figure 3.1.11 and 3.1.12.

Length Proportion: Based on our hypothesis that a basement is a combination of the LPE and the UPE with $N(S_{LPE}) = N(S_{UPE})$, $n_t(S_{LPE}) = n_t(S_{UPE})$ with $t \in \{a, d, e\}$ and $n_b(S_{LPE}) = n_f(S_{UPE})$, we expect that the length of a module t_i and its inverted module, $t_{N-(i-1)} \,|\, \forall i = 1,...,N$ in the basement module sequence $S_{basement} = (t_1,...,t_i,...,t_N)$, are equivalent: $l_{t_i} = l_{t_{N-(i-1)}}$. We define such property as *Scale Symmetric Property* (SSP). The length functions of ${}_1a, b, (e/{}_2a$ with $n_a = 3), f$ and ${}_{n_a}a$ shown on the left hand side of basement number 03, 05, 07, 08, and 09 in figures 3.1.11 and 3.1.12 however, give us different information. But still in cases such as basement 01, 02, 04 and 06 we can observe that $l_{{}_1a} \cong l_{{}_2a}$ and $l_b \cong l_f$ which satisfy our expectation. Extreme deviations parametric data from our hypothesis can be seen from basements number 03 and 05 where the lengths of all five module types occur arbitrary. Module e or ${}_2a$ is a central module, and therefore does not influence the symmetry.

The ratios of the lengths of b, e, f and ${}_2a$ are comparatively simple. We know from the general condition that the number of these groups are limited to one, and $n_a \leq 3$. In any case no more then five modules appear, and moreover, in small scale basements e and/or b and f can even vanish. In order to define the ratios of ideal lengths of b, e, f and ${}_2a$ we calculate the average length $\tilde{l}_t$ for $t = \{b, e, f, {}_2a\}$ using the captured data of the basements shown in figure 3.1.11 and 3.1.12, we obtain:

$$\tilde{l}_b = \frac{\sum_{i=1}^{8} l_{b_i}}{8} = 1,73,\ \tilde{l}_f = \frac{\sum_{i=1}^{8} l_{f_i}}{8} = 1,71,\ \tilde{l}_e = \frac{\sum_{i=1}^{9} l_{e_i}}{9} = 1,17,\ \tilde{l}_{{}_2a} = \frac{\sum_{i=1}^{9} l_{({}_2a)_i}}{9} = 0,98$$

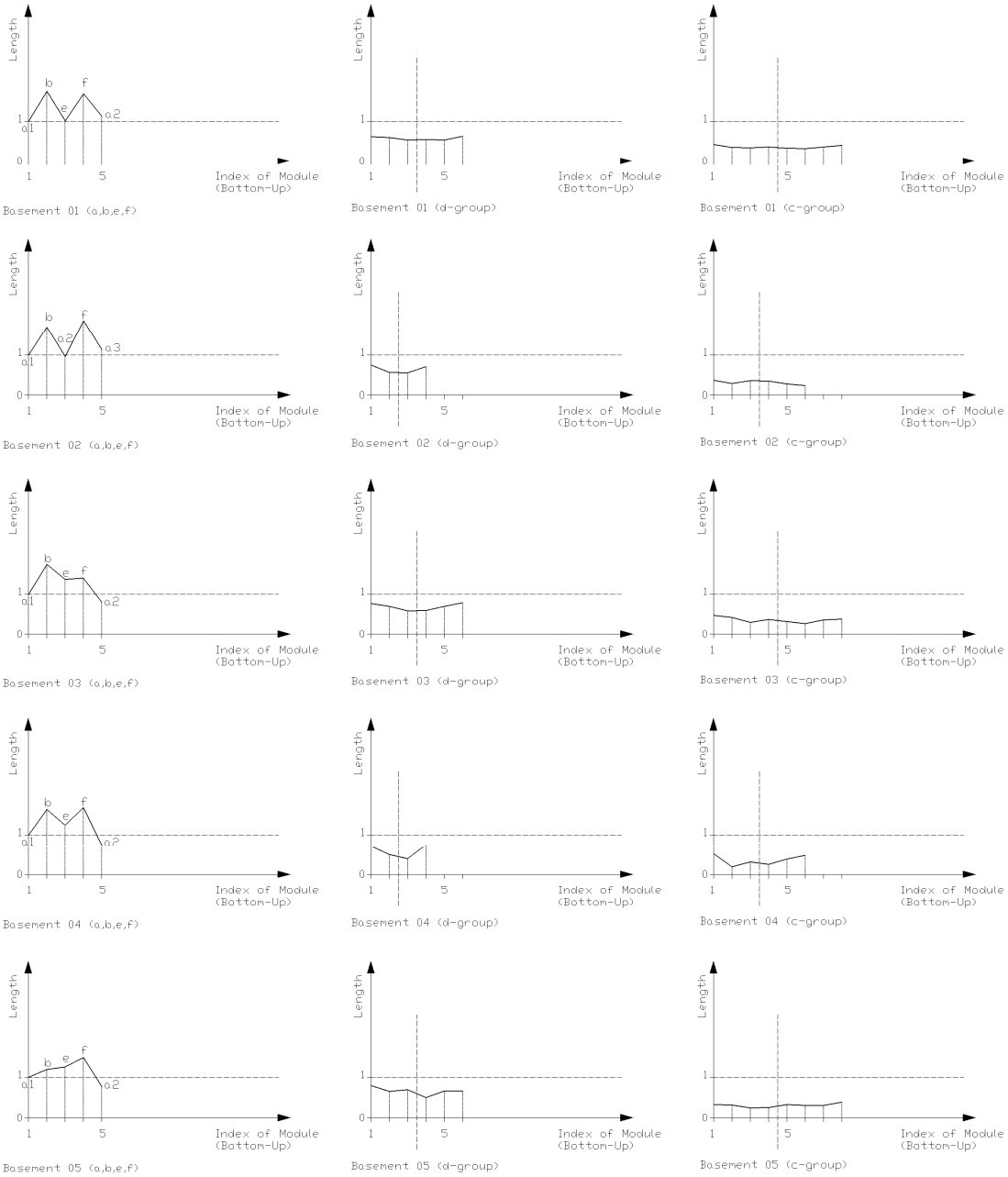


Figure 3.1.11
Graphs of captured lengths of modules in five different basements of Angkor Wat, Chao Say Tevda and Thomannon temple. On the left: the ratio of *b,e,f* and $_2a$ to $_1a$, and in the middle and on the right: graphs of *d*- and *c*-group respectively.

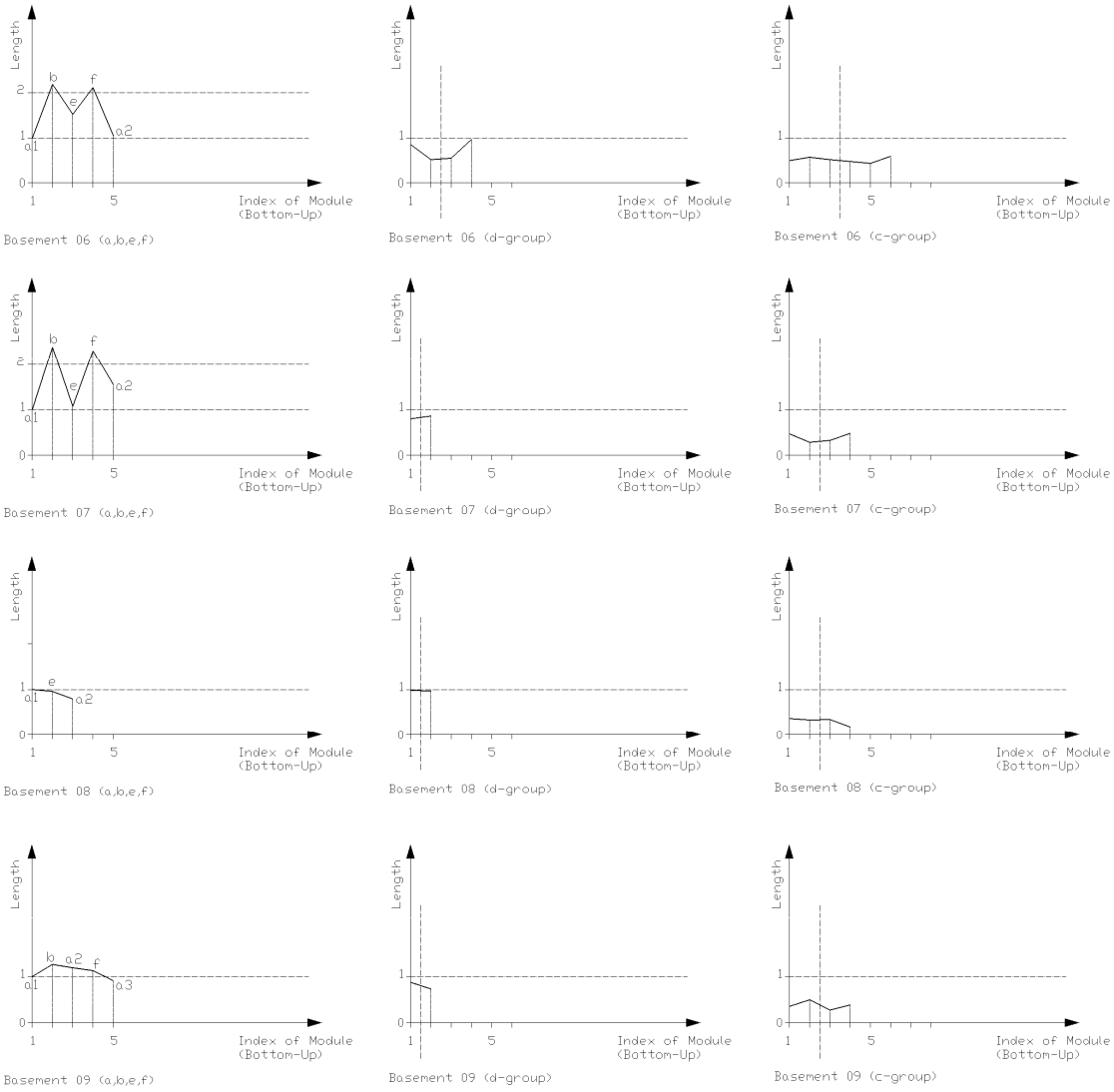


Figure 3.1.12
Graphs of captured lengths of modules in five different basements of Angkor Wat, Chao Say Tevda and Thomannon temple. On the left: the ratio of *b,e,f* and $_2a$ to $_1a$, and in the middle and on the right: graphs of *d*- and *c*-group respectively.

Remark:

We limit the input data for analysis to these nine basements however, larger dataset might provide better results. An interesting approach is also to analyze the proportion of modules based on distinct complexity of the element.

It is remarkable that the results strongly support our theory. We obtain the ideal lengths $l_{1a} \cong l_{2a} \cong 1$ and $l_b \cong l_f \cong 1,72$, and give an almost perfect symmetric graph of length proportion shown in figure 3.1.13.

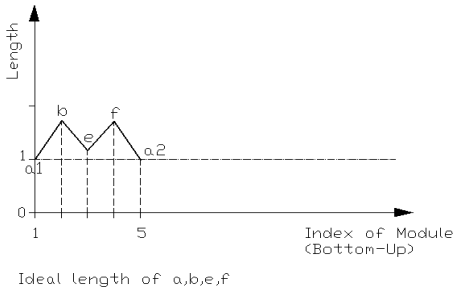


Figure 3.1.13
Graph of the average length of b, e, f and $_2a$.

Turing to the *c*- and *d*-modules we need a different way to represent the length ratios. Moreover, we sometimes observe quite a high number of the modules. The reason for this is that we require an *ideal function* to represent the length of each set of module. We again expect that for both the *d*- and the *c*-group to satisfy SSP. From the graphs we can see that nine of the ten selected cases including basement number 01, 02, 03 and 07 can be considered to adhere to this property with only minor artistic deviation.

We can observe for basements with a high number of repeated modules $(n_d > 2)$ that the curves representing the length of *d* all tend to have U-shaped curves (most noticeable 01, 02, 03, 04 and 06). More variations occur with the group of *c*. Here three kinds of curves exist: the upside down U-shaped curve (02 and 08), the W-shaped curve (01, 03 and 05) and the U-shaped curve (06 and 07).

The largest recorded numbers of *c*- and *d*-module on existing temples are $n_c = 8$ and $n_d = 6$ which can be found for instance on the main basement of the central building of Angkor Wat. However our vision is to open possibilities for ATE profile reconstructions with more complex structure and a larger number of *d*-modules. We can observe that the length function of the *d*- and even more the *c*-modules are not linear, which can be seen from basement 01, 02, 03, 05 and 06. These properties are related to Khmer arts and architectural means of smoothness we have mentioned in chapter 2. Thus

the function of the ideal lengths of *c*- and *d*-modules should be a curve. We suggest the cosine function as the length function of *c* and *d*-modules.

We define:

$$f(x) = A \cdot \cos x + B \tag{3.4}$$

as a shifted cosine function with radius A and shift parameter B. Additionally let us define Δl_t be the difference of maximum and minimum length of modules t in $S_{basement}$ thus $\Delta l_t = l_t^{\max} - l_t^{\min}$ for $t = \{c, d\}$. To calculate A and B from a given set of data, first observe that $\Delta l = l^{\max} - l^{\min}$ we get:

$$A = \frac{\Delta l}{2} \text{ and } B = \frac{\Delta l}{2} + l^{\min}.$$

In our dataset a small deviation from the shape sequence property occur. To calculate $l_c^{\max}$ and $l_d^{\max}$ respectively, we use an average over the first and last module length:

$$l_t^{\max} = \frac{\tilde{l}_{1t} + \tilde{l}_{(n_t t)}}{2}$$

where $\tilde{l}_{1t}$ and $\tilde{l}_{(n_t t)}$ denoting the average lengths of the first and the last module of type t respectively. We get the following

$$\tilde{l}_{1c} = \frac{\sum_{i=1}^{9} l_{(1c)_i}}{9} = 0,416; \quad \tilde{l}_{(n_c c)} = \frac{\sum_{i=1}^{9} l_{(n_c c)_i}}{9} = 0,386 \quad \Rightarrow l_c^{\max} = \frac{\tilde{l}_{1c} + \tilde{l}_{(n_c c)}}{2} = 0,401$$

$$\tilde{l}_{1d} = \frac{\sum_{i=1}^{9} l_{(1d)_i}}{9} = 0,787; \quad \tilde{l}_{(n_d d)} = \frac{\sum_{i=1}^{9} l_{(n_d d)_i}}{9} = 0,778 \quad \Rightarrow l_d^{\max} = \frac{\tilde{l}_{1d} + \tilde{l}_{(n_d d)}}{2} = 0,782.$$

For $l_c^{\min}$ and $l_d^{\min}$ respectively, we use an average over the minimal module length that occurs in our dataset:

$$l_c^{\min} = \frac{\sum_{i=1}^{9} l_{c,i}^{\min}}{9} = 0,266; \quad l_d^{\min} = \frac{\sum_{i=1}^{9} l_{d,i}^{\min}}{9} = 0,615$$

With these values we now compute:

$$\Delta l_c = l_c^{\max} - l_c^{\min} = 0,401 - 0,266 = 0,135; \quad \Delta l_d = 0,782 - 0,615 = 0,167.$$

The ideal lengths function of *d*-module groups for instance can be obtained by firstly calculating:

$$A = \frac{\Delta l_d}{2} = \frac{0{,}167}{2} = 0{,}084; \quad \Rightarrow B = 0{,}084 + 0{,}615 = 0{,}699 \text{, and further}$$

$$f_d(x) = 0{,}084 \cdot \cos x + 0{,}699 \qquad (3.5)$$

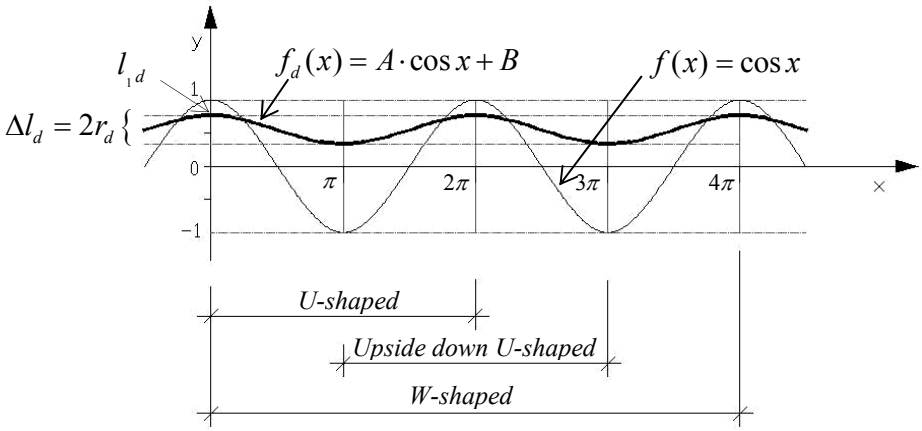


Figure 3.1.14
Cosine function: representation of the *d*-module length function.

Similarly, the ideal lengths function of *c*-group:

$$A = \frac{\Delta l_c}{2} = \frac{0{,}135}{2} = 0{,}068; \quad \Rightarrow B = 0{,}068 + 0{,}266 = 0{,}334 \text{, and the function}$$

$$f_c(x) = 0{,}068 \cdot \cos x + 0{,}334 \qquad (3.6)$$

The advantage of the function $f_t(x)$ is that it represents all three different shapes of length functions of *c*- and *d*-module groups that we have introduced above. Each shape is defined by different interval $\left\{x \middle| x_1 \le x \le x_{n_t}\right\}$ of $f_t(x)$ (see figure 3.1.14) where x_1 and x_{n_t} denoting the left-most and right-most value of *p* respectively, and can be described as following:

$f_t(x) \mapsto \cup$-shape for $x \in [0, 2\pi]$

$f_t(x) \mapsto \cap$-shape for $x \in [\pi, 3\pi]$

$f_t(x) \mapsto W$-shape for $x \in [0, 4\pi]$

The length $l_{i\,t} = f_t(x_i)$ of a given module t_i is defined by the position x_i within the certain interval $\left\{x \middle| x_1 \le x \le x_{n_t}\right\}$ for $i = 1,2,...,n_t$. We define $x_1 \mapsto f(x) = l_{1\,t}$ and $x_{n_t} \mapsto f(x) = l_{n_t\,t}$ to be the values that satisfy $f(x)$ with the length of the first and the last module of type *t* respectively. The length $l_{i\,t}$ of the remaining modules $t_i \middle| \forall i \ne 1, n_t$ is the value $x_i \mapsto f(x) = l_{i\,t}$ with a constant distance $\Delta x = x_{i+1} - x_i$ between x_i and x_{i+1} in order to obtain a smooth transition of lengths $l_{i\,t}$ and $l_{i+1\,t}$, and can be formulated as following:

$$\Delta x = \frac{x_{n_t} - x_1}{n_t - 1}. \qquad (3.7)$$

With (3.7) we calculate: $x_i = x_1 + \Delta x(i-1)$ (3.8)

Until this step we know the length l_t of the all the modules (module vectors) in a basement element but yet we do not know their angle α_t. The following step we will define the angle α_t of *t*-module in a module sequence $S_{basement}$ of a basement.

Angle Proportion: Similarly to the method of capturing the lengths of modules in a basement we have seen earlier, we are now trying to understand the angle proportions of modules in an Angkor temple basement. We select nine basements to capture the angle value of each module it consists of, and observe its angle development as we switch from one module to another step-wise upward. Again we observe them by the same three different groups as we have done to the height capture. The results are depicted in a graph as shown in figure 3.1.15 and figure 3.1.16. Immediately we remark the almost identical property of the graph of the group *a, b, e* and *f* as shown on left hand side graph. Remarkable is also the angle value of the initiator ${}_1a$ and the value of the central module which are in these cases ${}_2a$ or e. Here we obtain $\alpha_t \cong \frac{\pi}{2} \middle| \forall t = \{a, e\}$. Module *b* and *f* on the other hand appear to be a reflection of each other which angles tend toward a relation of $\alpha_f = \pi - \alpha_b$.

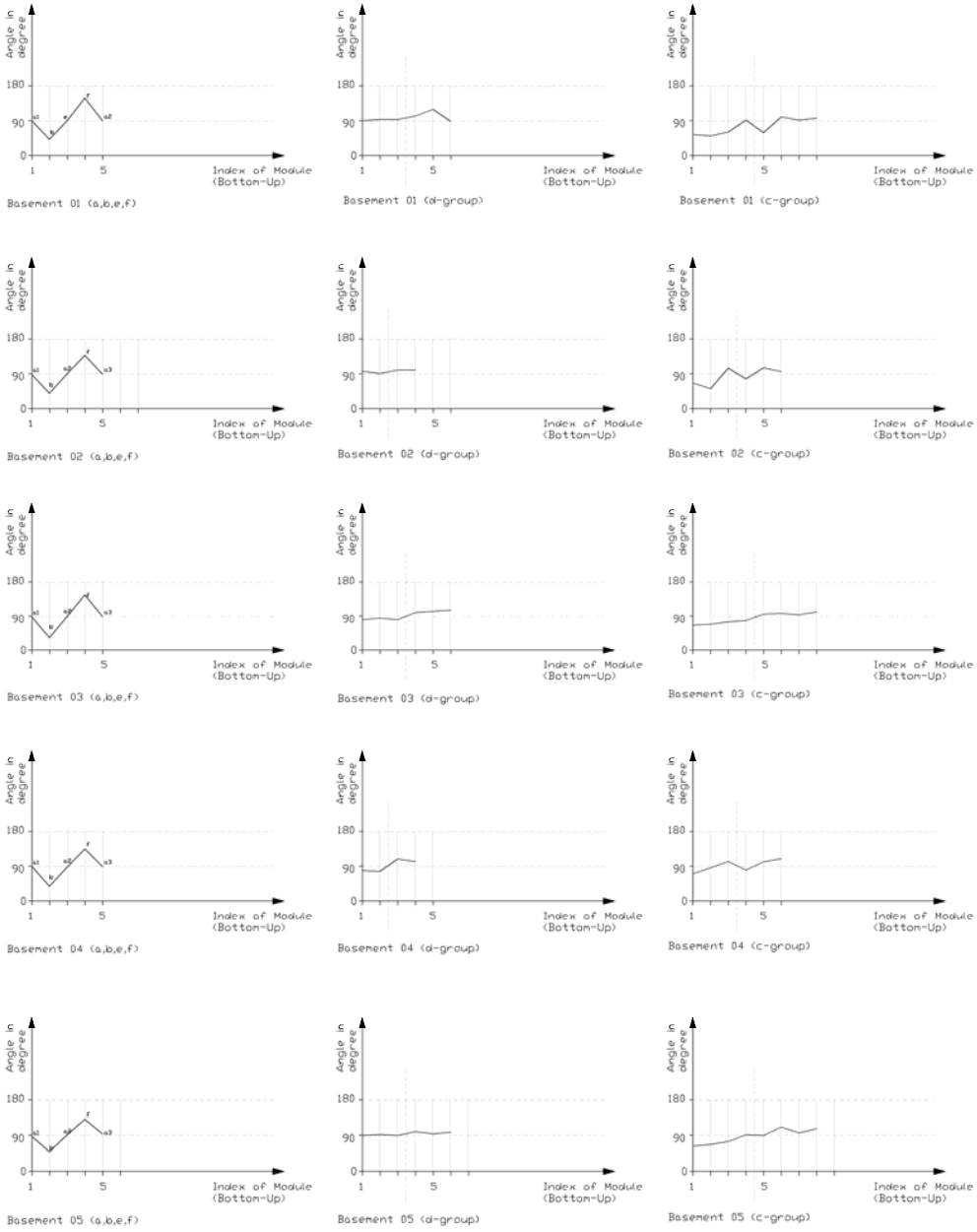


Figure 3.1.15
Angle proportions of modules in five selected Angkor temple basements. On the left: compare the proportions of *a, b, e* and *f,* in the middle within *d*-group, and on the right within *c*-group.

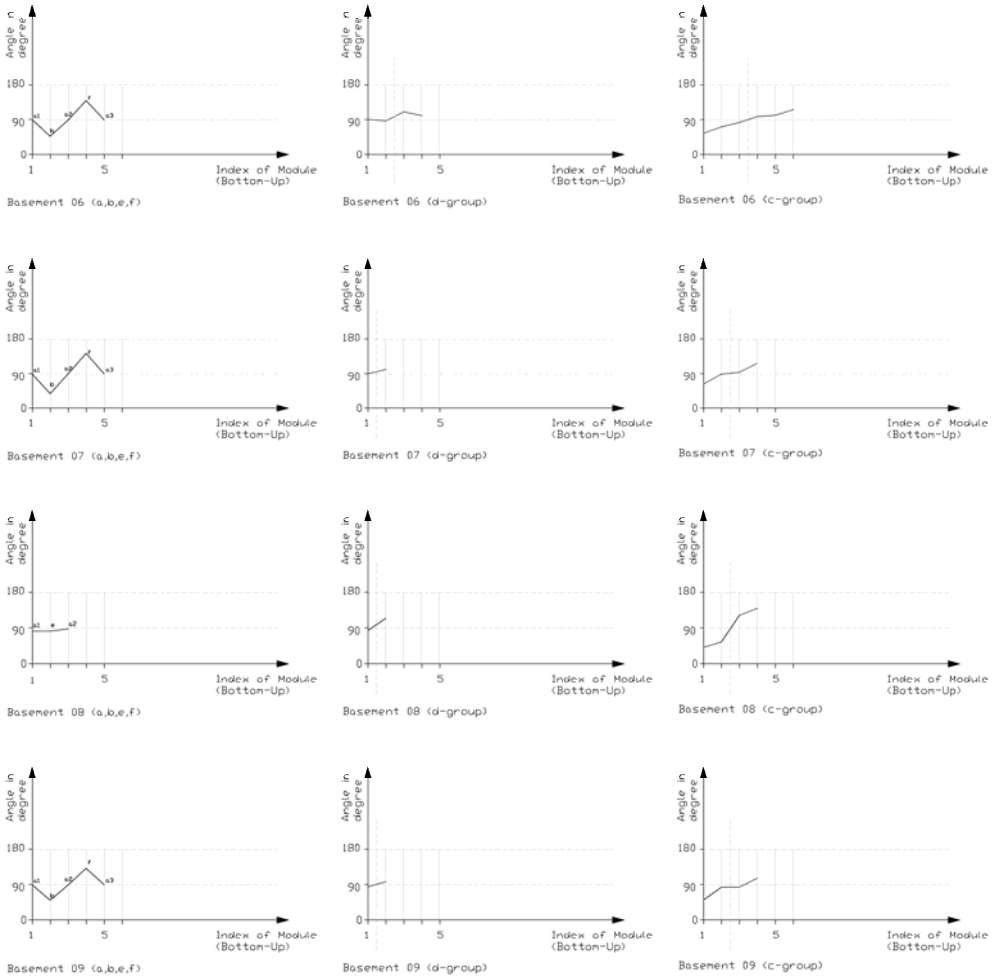


Figure 3.1.16
Angle proportions of modules in five selected Angkor temple basements. On the left: compare the proportions of *a, b, e* and *f,* in the middle within *d*-group, and on the right within *c*-group.

Let $\tilde{\alpha}_b$ and $\tilde{\alpha}_f$ be the average angle of *b* and *f* respectively. We calculate the ideal angle $\alpha_b = \tilde{\alpha}_b$ and $\alpha_f = \tilde{\alpha}_f$ and obtain:

$$\alpha_b = \tilde{\alpha}_b = \frac{\sum_{i=1}^{8} \alpha_{b_i}}{8} = \frac{338{,}14^\circ}{8} = 42{,}27^\circ \cong \frac{7}{30}\pi$$

$$\alpha_f = \tilde{\alpha}_f = \frac{\sum_{i=1}^{8} \alpha_{f_i}}{8} = \frac{1108{,}82^\circ}{8} = 138{,}6^\circ \cong \frac{23}{30}\pi .$$

We see that indeed $\frac{23}{30}\pi = \pi - \frac{7}{30}\pi$. Thus we define value $\alpha_b = \frac{7}{30}\pi$ and $\alpha_f = \frac{23}{30}\pi$ to be the ideal angles of t_b and t_f respectively.

The graphs of angles developments in c-group and d-group have a common characteristic. Here we obtain almost a continuous rising of the angle with a drastic change of gradient at the crossing part between the lower part and the upper part modules of the basement (crossing part is defined by the vertical dashed lines in graphs of *c*- and *d*-group). The domain of angle changes of ATE modules is $0 < \alpha_t < \pi$ for $t = \{a, b, c, d, e, f\}$.

The lowest recorded angle value from our input data is around 32 degree belongs to *b*-module from basement 03 whereas the highest is around 148 degree of *f*-module from basement 01. Here again we see that $\alpha_f = \pi - \alpha_b$. These relationships we also expect from group *d* and *c* individually thus $\alpha_{t_{(N-1)-i}} = \pi - \alpha_{t_i}$ for $t = \{c, d\}$. The numerical data, however, shows that there is no any remarkable common rule for controlling the angles of both module types. Nevertheless there still exit some pairs of modules in the same basement that satisfy $\alpha_{t_{(N-1)-i}} = \pi - \alpha_{t_i}$ with ± 5 degree tolerance. These special cases is marked by a star symbol (*) next to the angle values in column $\pi - \alpha_{t_{n_t+1-i}}$ which is compared with α_{t_i} for $t = \{c, d\}$ and $i = 1, 2, ..., n_t$ (see table of appendix A.2).

Based on this information we could make a conclusion that an ideal Angkor temple basement with module sequence $S_{basement}$ generally comprises of modules of which the angle value α_{t_i} and $\alpha_{t_{(N-1)-i}}$ of the pair t_i and $t_{(N-1)-i}$ respectively has the following proportion:

$$\alpha_{t_{(N-1)-i}} = \pi - \alpha_{t_i} \text{ for } t = \{c, d\} \text{ and } i = 1, 2, ..., N . \qquad (3.7)$$

The next step we need to know the initial angle value of the module to be initialized at first in the module sequence, and the angle difference $\alpha_{i+1\,t} - \alpha_{i\,t}$ of two adjacent modules ${}_{i}t$ and ${}_{i+1}t$ so that all modules of type $t = \{c, d\}$ can be generated based on a certain ideal angle function of ATE modules.

From the captured data we have presented in figure 3.1.15 and 3.1.16 we can observe that the initial angle value of *c*-group varies between 41 and 69 degrees, and that of *d*-group between 81 and 95 degrees. The same method we have used to calculate the ideal initial length of both module groups will be applied to define the ideal initial angle $\alpha_{1\,t}$ for $t = \{c, d\}$ as well. Let $\tilde{\alpha}_{1\,t}$ be the average angle value of the initial module type $t = \{c, d\}$ of all nine selected basements. The ideal initial angle is calculated by $\alpha_{1\,t} = \tilde{\alpha}_{1\,t} = \frac{\sum_{i=1}^{9} \alpha_{(1\,t)_i}}{9}$ where *i* denoting the index of the basement. We obtain the ideal initial angle of *c* and *d* (see also appendix A.2):

$$\alpha_{1\,c} = \tilde{\alpha}_{1\,c} = \frac{\sum_{i=1}^{9} \alpha_{(1\,c)_i}}{9} = 58,66^{\circ}, \text{ and} \qquad (3.8)$$

$$\alpha_{1\,d} = \tilde{\alpha}_{1\,d} = \frac{\sum_{i=1}^{9} \alpha_{(1\,d)_i}}{9} = 86,88^{\circ}. \qquad (3.9)$$

From (3.7), (3.8) and (3.9) we can calculate

$$\alpha_{n_t\,t} = \pi - \alpha_{1\,t} \qquad (3.10)$$

and thus we know the difference $\alpha_{n_t\,t} - \alpha_{1\,t}$. With the given number n_t of module *t* we can further calculate the difference of $\alpha_{i\,t}$ and $\alpha_{i+1\,t}$. Let $\Delta\alpha_t = \alpha_{i+1\,t} - \alpha_{i\,t}$ be the difference of $\alpha_{i\,t}$ and $\alpha_{i+1\,t}$. We then obtain:

$$\Delta\alpha_t = \frac{\alpha_{n_t\,t} - \alpha_{1\,t}}{n_t - 1} \qquad (3.11).$$

We replace (3.10) in (3.11), and get:

$$\Delta\alpha_t = \frac{\pi - 2\alpha_{1t}}{n_t - 1} \qquad (3.12).$$

Finally we obtain the angle α_{it} at *i*-th index as following:

$$\alpha_{it} = \alpha_{1t} + (i-1)\Delta\alpha_t = \alpha_{1t} + (i-1)\frac{\pi - 2\alpha_{1t}}{n_t - 1} = \frac{\alpha_{1t}(n_t - 1) + (i-1)(\pi - 2\alpha_{1t})}{n_t - 1}$$

$$\Rightarrow \alpha_{it} = \frac{\pi(i-1) + \alpha_{1t}(n_t - 2i + 1)}{n_t - 1} \qquad (3.13)$$

Summary: Module Proportions of $S_{basement}$

Module Length: The proportional setting of modules t_i in an Angkor temple basement $S_{basement} = (t_1,...,t_i,...,t_N)$ for $t = \{a,b,c,d,e,f\}$ and $i = 1,...,N$ is divided into two steps: firstly we set the length l_{t_i} and secondly rotate t_i by α_{t_i}.

We define the ideal length of *a, b, e* and *f* by $l_b = l_f = 1,72$ and $l_e = 1,17$ respectively for $n_a = 2$ and $\{n_b, n_e, n_f\} \neq 0$. The ideal lengths of module groups of *c* and *d* are l_{it} with $t = \{c,d\}$ for $i = 1,...,n_t$ is defined by the function:

$$f_t(x) = A \cdot \cos x + B$$

with A: radius of length unit circle

ideal radius: $A_c = 0,068;\ A_d = 0,084$

B: Shifting Parameter

$B_c = 0,334;\ B_d = 0,699.$

x_i: position of module index *i* that satisfy $f(x) = l_{it}$ and is calculated as:

$$x_i = x_1 + \Delta x(i-1) \text{ for } i = 1,...,n_t \text{ and } \Delta x = \frac{x_{n_t} - x_1}{n_t - 1}.$$

Function $f_t(x)$ represents all three different shape of length function which can be distinguished by the period $p = \left[x_1, x_{n_t}\right]$.

Module Angle: The rotation of module t_i by an ideal angle α_{t_i} is again separated into two steps. Firstly we define α_{t_i} for $t=\{a,b,e,f\}$: $\alpha_a=\alpha_{1a}=\alpha_{2a}=\alpha_e=\frac{\pi}{2}$; $\alpha_b=\frac{7}{30}\pi$ and $\alpha_f=\pi-\alpha_b$. Then we rotate t_i for $t=\{c,d\}$ and $i=1,...,n_t$ using the following formula:

$$\alpha_{it}=\frac{(i-1)\pi+\alpha_{1t}(n_t-2i+1)}{n_t-1}$$

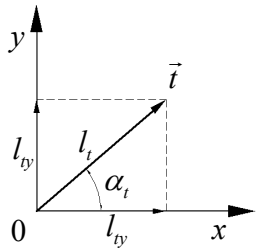
with α_{1t}: initial angle

ideal $\alpha_{1c}=\frac{59}{180}\pi$; $\alpha_{1d}=\frac{87}{180}\pi$

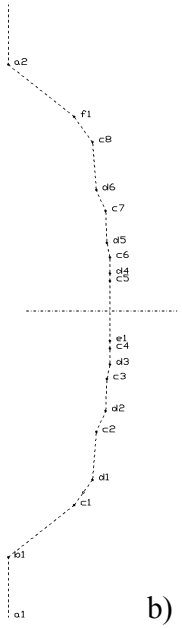
n_t: total number of input module type *t*

After the proportional settings of length and angle to each module (module vector) t_i of $S_{basement}=(t_1,...,t_i,...,t_N)$, the structure of *S* is no longer a composition of homogeneous line segment, and its geometry is not a simple horizontal line anymore. We can now describe *S* in form of vector as following (see also figure 3.1.17):

$$\begin{aligned}S_{basement}&=\left(\vec{t}_1,...,\vec{t}_i,...,\vec{t}_N\right)\\&=\left(\begin{pmatrix}l_{t_1y}\\l_{t_1x}\end{pmatrix},...,\begin{pmatrix}l_{t_iy}\\l_{t_ix}\end{pmatrix},...,\begin{pmatrix}l_{t_Ny}\\l_{t_Nx}\end{pmatrix}\right)\\&=\left(l_{t_1}\begin{pmatrix}\cos\alpha_{t_1}\\\sin\alpha_{t_1}\end{pmatrix},...,l_{t_i}\begin{pmatrix}\cos\alpha_{t_i}\\\sin\alpha_{t_i}\end{pmatrix},...,l_{t_N}\begin{pmatrix}\cos\alpha_{t_N}\\\sin\alpha_{t_N}\end{pmatrix}\right)\end{aligned}$$



(a)



b)

Figure 3.1.17
(a) Vector representation of *t* with magnitude l_t and rotated by α_t.
(b) Reconstruction of a basement structure $S_{basement}$ with $n_a=2$, $n_b=1$, $n_d=6$, $n_e=1$ and $n_f=1$.

LPEs and UPEs

The proportional controls of both, module lengths and angle, we have presented so far is specified for reconstruction of Angkor temple basement. LPEs and UPEs have their own characteristics which must be handled differently. However, referring to the theory of the architectonic relationship of the basement with a LPE and an UPE (see section 2.2.3; figure 2.2.9), we can partially apply information for reconstruction of basement structure to reconstruct LPEs and UPEs.

The proportional property of modules in the sequence $S_{LPE} = (t_1,...,t_i,...,t_N)$ of a LPE is identical to the lower-part segment $S'_{basement} = \left(t_1,...,t_i,...,t_{\frac{M'-1}{2}}\right)$ of a basement $S_{basement}$ with $t = \{a,b,c,d\}$ and $f_t(x) \mapsto \cup$-shaped curve. A special property of the LPE structure is that $l_{t_i} > l_{t_{i+1}}$ with $t = \{c,d\} \bigcup \{{}_2a, e \in t(d)\}$ for $i = 1,...,n_t$ considering the structure upward. The notion ${}_2a, e \in t(d)$ notice that we include the module type *a* and *e* into *d*-group for the case of using the function $f_t(x)$ to calculate the lengths of $t = d$. Thus we can conclude that the length function of *t* in a LPE is a decreasing monotony function. With these prerequisite we can conclude:

1. The ideal lengths of modules in LPE is controlled as following:
   - The ideal lengths of ${}_1a$ and b are identical to that of a basement: $l_{{}_1a} = 1,\ l_b = 1,72$.
   - Groups *c* and *d* can be controlled by the function $y = f_t(x) = r_t(\cos x - 1) + l_{{}_1t}$ within an interval $\{x | 0 \le x \le \pi\}$ and $x_1 \mapsto 0;\ x_{n_t} \mapsto \pi$, $t = \{c,d\} \bigcup \{{}_2a, e \in t(d)\}$. Again we define $y = f_t(x_i) = l_{{}_it}$ where x_i can be calculated by $x_i = x_1 + \Delta x(i-1)$ for $i = 1,...,n_t$ and $\Delta x = \dfrac{x_{n_t} - x_1}{n_t - 1}$.

- The initial lengths l_{1t} for $t=\{c,d\}$ are identical to that of a basement: $l_{1c}=0,401$ with $r_c=0,167$; $l_{1d}=0,782$ with $r_d=0,345$.

2. Angle proportions can be controlled in the same way as for a basement:

- ideal angle $\alpha_{1a}=\alpha_e=\frac{\pi}{2}$; $\alpha_b=\frac{7}{30}\pi$; $\alpha_{2a}=\frac{5}{12}\pi$ (see appendix B.1).
- Angle α_{it} can be calculated using:

$$\alpha_{it}=\frac{\frac{\pi}{2}(i-1)+\alpha_{1t}(n_t-i)}{n_t-1} \qquad (3.14)$$

for $t=\{c,d\}\cup\{e\in t(d)\}$ and initial $\alpha_{1c}=\frac{59}{180}\pi$, $\alpha_{1d}=\frac{87}{180}\pi$. See appendix B.2 for details.

On the contrary, the structure $S_{UPE}=(t_1,...,t_i,...,t_N)$ of UPE comprise of modules t_i of which lengths $l_{t_i}<l_{t_{i+1}}$ for $i=1,...,n_t$ and $t=\{c,d\}\cup\{{}_1a,e\in t(d)\}$, and thus forming an increasing monotony function $f_t(x)$. The calculation of ideal lengths l_{it} is done following the procedure we presented in step 1 for computing the lengths of modules in LPE above. Different is the interval of x which is specified for UPE: $\{x|\pi\le x\le 2\pi\}$. We see that the length l_{1a} of the initial module ${}_1a$ is $l_{1a}<1$, and ${}_1a$ is classified in the group of d. So we can calculate l_{1a} from function (3.5) with $x_1=\pi$:

$$l_{1a}=f_d(x_1)=r_d(\cos x-1)+l_{1d}=0,177(\cos\pi-1)+0,782$$
$$\Rightarrow l_{1a}=0,428$$

Lengths l_{it} for $i=\{1,...,n_t\}$ and $t=\{c,d\}\cup\{e\in t(d)\}$ can be done in the same way as for LPE using function (3.5) and (3.6). The rest, l_f and l_{2a} remains the same as we have defined for a basement.

Angle proportion of UPE modules is identical to the modules of the upper part segment of a basement. The initial ideal angle is $\alpha_{1a} = \pi - \frac{5}{12}\pi = \frac{7}{12}\pi$ (see appendix B.1) which is followed by α_{it} with $t = \{e, d\}$ for $i = 1,...,(n_d + n_e)$, and can also be calculated using the equation (3.14). The initial module of *d*-group is *e* with an angle $\alpha_e = \frac{\pi}{2}$ in case $n_e \neq 0$.

Summary: We are approaching the final step to complete the reconstruction of the ideal 2D profile of an ATE. Before we continue we would like to summaries at this point that the reconstruction of Angkor temple elements that is based on the number of input components is about reconstructing the sequence of their module vectors following a certain combination rule that is specified for a certain element type. After obtaining the proper module sequence, we applied the required proportion to each vector in order to form the horizontal line segments into a proper shape of an element as shown for a sample in figure 3.1.17, image (b). Yet the result is not a profile of an ATE. We still need to map the module profiles to the sequence, and is discussed in the following section.

3.1.4 Mapping the Module Profiles

We have introduced in section 2.3.1 and definition 2.3.2 that a module profile p_t of module type *t* is a non-closed sequence of edge segments. We also know that it consists of two ends: a start point s^{p_t} and end point e^{P_t}. The same as a module vector *t*, a module profile $p_{0,t}$ is initialized at the origin of the coordinate system (*x, y*): with $s^{p_{0,t}}(0,0)$ and $e^{p_{0,t}}(1,0)$. Our goal is to map the start point $s^{p_t} \mapsto s_t$ and $e^{p_t} \mapsto e_t$ with $\{s^{p_t}, e^{p_t}\} \in p_t$ and $\{s_t, e_t\} \in t$. To do this we need to proceed three steps 2D transformations.

Firstly we will fit the dimension of an initialized $p_{0,t}$ to the dimension of *t*, thus fit to the length $l_t = (l_{(x)}, l_{(y)})$ of *t*, by multiplying each node $\{\sigma_i'''(x_i''', y_i''') \in \mathbb{R}^2\} \in p_{0,t}$ with

the scaling matrix ς_l for $i = 1,...,n$ and $t = \{a,b,c,d,e,f\}$ where *n* denotes the number of nodes that form p_t. We obtain $\{\sigma_i''(x_i'', y_i'') \in \mathbb{R}^2\} \in p_t''$:

$$\sigma_i''(x_i'', y_i'') = \varsigma_l \sigma_i''' = \begin{bmatrix} l_{(x)} & 0 \\ 0 & l_{(y)} \end{bmatrix} \begin{bmatrix} x_i''' \\ y_i''' \end{bmatrix} = \begin{bmatrix} l_{(x)} x_i''' \\ l_{(y)} y_i''' \end{bmatrix}. \tag{3.15}$$

We transform $p_{0,t}$ to p_t'' by keeping the ratio of length and width of p_t, thus $l_{(x)} = l_{(y)} = |l_t|$.

Secondly we need to rotate p_t'' such that vector $(s^{p_t''}, e^{p_t''})$ points to the same direction as (s_t, e_t). In other words, we need to rotate $\sigma_i''(x_i'', y_i'')$ in counterclockwise direction by α_t over the origin $(0,0)$ for $i = 1,...,n$, where α_t denotes the angle of module vector *t* with the *x*-axis. Let $\sigma_i'(x_i', y_i') \in \mathbb{R}^2$ be the nodes of p_t' after rotating p_t'' over the start point $s^{p_t''}(0,0)$. We obtain $\sigma_i' \in p_t'$:

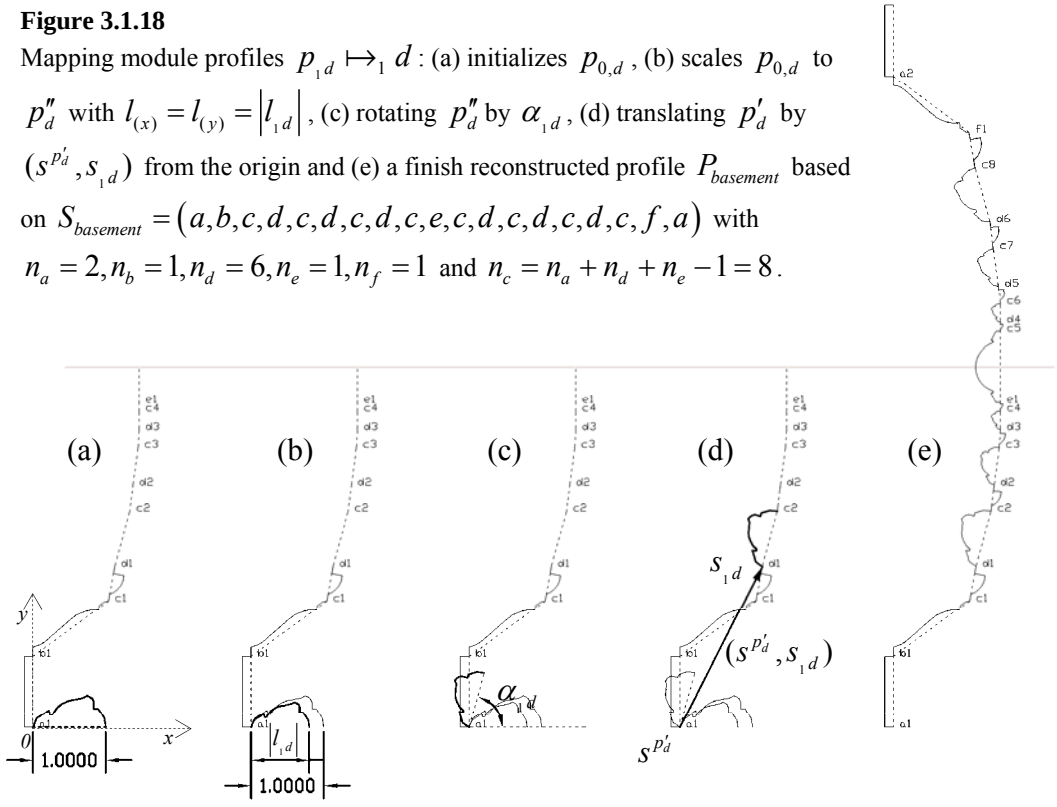
$$\sigma_i' := \left\{ \begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} \cos\alpha_t & -\sin\alpha_t \\ +\sin\alpha_t & \cos\alpha_t \end{bmatrix} \begin{bmatrix} x_i'' \\ y_i'' \end{bmatrix}. \right. \tag{3.16}$$

Finally in step three we need to move p_t' to map to *t*, and define p_t as the final stadium of the transformation. The mapping is done in such a way that $s^{p_t} \mapsto s_t$ thus we translate p_t' along the vector (s^{p_t}, s_t). Each node $\{\sigma_i(x_i, y_i) \in p_t\} \in \mathbb{R}^2$ for $i = 1,...,n$ can be found using:

$$\sigma_i := \left\{ \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_i' \\ y_i' \end{bmatrix} + \begin{bmatrix} x_{s_t} \\ y_{s_t} \end{bmatrix}. \right. \tag{3.17}$$

In figure 3.1.17 shows a graphics representation of the transformation procedure demonstrating a map $p_{1d} \mapsto_1 d$. After all profiles are mapped we obtain an ideal 2D basement profile as shown in image (e) of the figure 3.1.18, and is ready for 3D reconstruction which will be discussed in the last section 3.4.

Figure 3.1.18
Mapping module profiles $p_{1\,d} \mapsto_1 d$: (a) initializes $p_{0,d}$, (b) scales $p_{0,d}$ to p''_d with $l_{(x)} = l_{(y)} = \left| l_{1\,d} \right|$, (c) rotating p''_d by $\alpha_{1\,d}$, (d) translating p'_d by $(s^{p'_d}, s_{1\,d})$ from the origin and (e) a finish reconstructed profile $P_{basement}$ based on $S_{basement} = (a,b,c,d,c,d,c,d,c,e,c,d,c,d,c,d,c,f,a)$ with $n_a = 2, n_b = 1, n_d = 6, n_e = 1, n_f = 1$ and $n_c = n_a + n_d + n_e - 1 = 8$.



3.2 Capturing the Element Structure from an Image

The method for reconstruction of Angkor temple element profile we have proposed in previous section is likely more suitable for reconstruction of ideal 2D element profiles following a pre-defined rule which potentially has been the traditional Khmer temple construction rule. In some cases however, we need to reconstruct profiles based on available input data of an actual existing temple element such as basic measurements of the height, width of the element and digital photography of it. The reason why we additionally propose this second method for reconstruction of ATE profiles is because in some cases (though seldom) we find elements of which module structure does not

correspond to the ATE group. Thus it is not possible to implement the principle rule for reconstruction of such exceptional structures. The image in figure 3.2.1 for example shows a wall base of Angkor Wat where a *b*-module, instead of being adjacent to *a*-module, it is directly connected to a *d*-module.

Figure 3.2.1
A special wall base structure: left, of the southern library of Angkor Wat's, and right, of Angkor Wat gopuram. Extraordinary location of *b* between *d*- and *c*-module in the sequence $S_{LPE} = (a,c,d,b,c,d,c,d,c,d,c,d,c,a)$.

In this section we are going to discuss on identification of module properties in an element structure. We mainly focus on identifying the type *t*, the start point s_t and the end point e_t of the module *t* as well as the way to capture their properties from the input image. Then we will describe an algorithm for reconstruction of the element structure, and conversion of the captured data in vector format of AutoCAD.

3.2.1 Identification of the Module Type

Our six predefined module types presented in section 2.3.1 do not always perfectly fit with the modules on actual elements. In figure 3.2.2 we present the major members of *e*-module family. There are two main criteria for identification and classification of the module type; the similarity of their profiles and the decorative aspect which is based on the same principle of lotus motif, e.g. image (1), (2) and (3) is one group of *e*, and (4), (5)

and (6) the other group of *e*. We can observe a certain similarity of the profiles of these modules if compared to other module types.

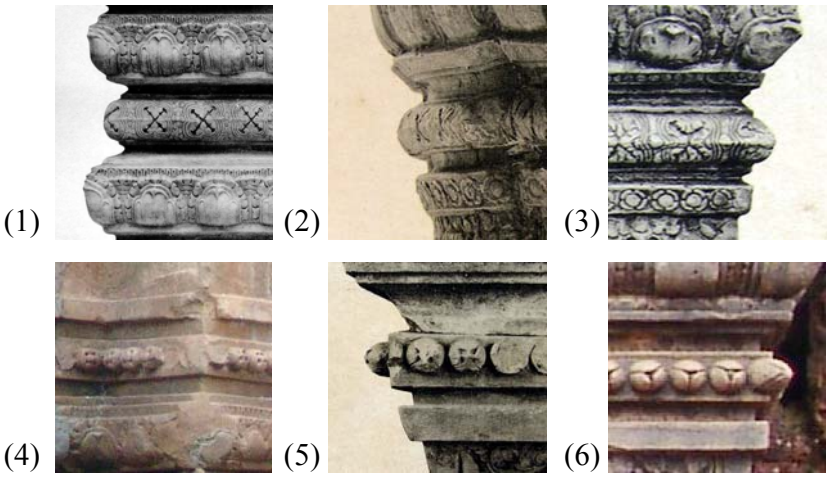


Figure 3.2.2
(1), (2) and (3) are respectively a basement , cylindrical column capital, and quadratic column capital of Angkor Wat depicting one sort of *e*-module. (4), (5) and (6) shows respectively a wall base of Chao Say Tevda, column capital of Boeung Mealea and column capital of Banteay Sray temple with another sort of *e*-module.

On the other hand, based on the fact that a man-made work is always unique, more over in art and architecture where favor and fantasy are an important factor for the creativity, we accept minor geometric deviations of modules in the same family. Our goal is to generate a virtual reality representation of temple elements using a limited complexity of their 3D geometry together. Advanced details of the modules (lotus motif) are expressed by the texture (photo of the element) to be mapped on the geometry. Figure 3.2.3 demonstrates the effectiveness of this method (see also appendix C). Here we see that even though we use the same 2D profile of a column capital to reconstruct two different styles of columns, we almost realize no anomaly of inappropriate match after applying texture to the geometry.

As the module type is identified, we know the position of its start point and end point which is relevant to calculate other properties such as length and angle of that module vector. In the following section we present the method to capture the module

vector of the identified module from an input image.

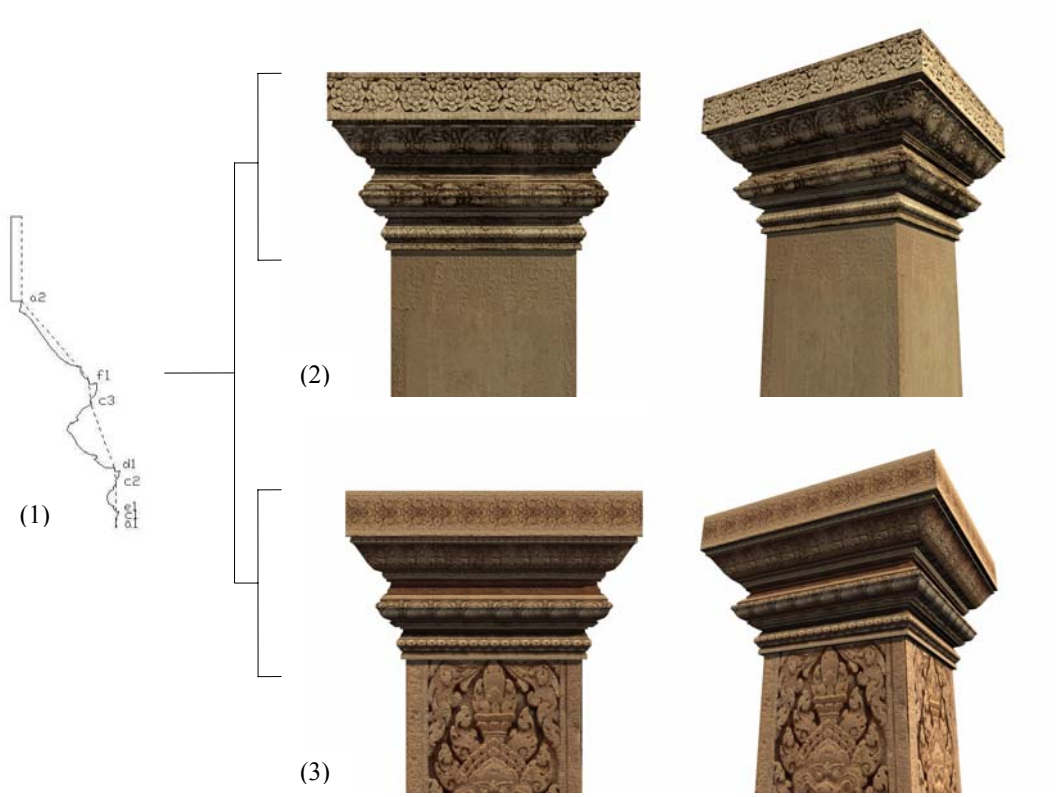


Figure 3.2.3
3D Computer reconstruction of column capital of Angkor Wat (2) and Banteay Srey (3) using two different textures (texture from figure 3.2.2 (3) and (6) respectively) but the same 2D geometry (1).

3.2.2 Capturing the Module Vectors

Before we discuss on capturing properties of modules from an image we need a mathematical notation of the digital image and its elements. Based on *Jähne* [Jäh02] an image is a grid $I_{\xi,\psi}$ with the pixel (element) $E(\xi,\psi)$ where *m* denoting the rows index and *n* the columns index of the grid, and can be represented by an $\Xi \times \Psi$-matrix. The start point *s* and end point *e* of the module vector $t(s,e)$ to be captured from the image *I* are denoted by two pixels $E_s(\xi_s,\psi_s)$ and $E_e(\xi_e,\psi_e)$ with $\psi_s > \psi_e$, see figure 3.2.4.

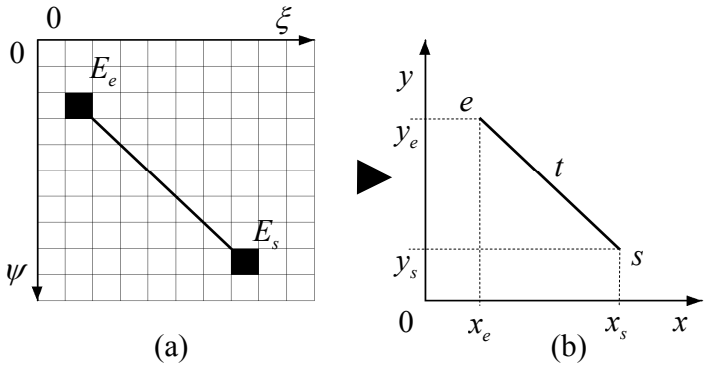


Figure 3.2.4
Conversion of coordinates from (ξ,ψ) of *I* to (x,y) of a CAD tool. (a) The grid of $I_{\xi,\psi}$ in which two pixels $E_s(\xi_s,\psi_s)$ and $E_e(\xi_e,\psi_e)$ are captured, and (b) shows the new coordinates of *t* with proportions as it was captured from *I* but in a CAD coordinate system.

We see that the column index ξ of *I* is arranged from left to right like the ordinate x of an ordinary coordinate system (x,y) but the row index (vertical scalar) ψ is pointing downward adapting the matrix notation. Related to this notation we define our module capturing process in top-down direction. In order to keep the proportion of *t* when vectorizing it in a conventional CAD coordinate system on the basis of the captured data $E_s(\xi_s,\psi_s)$ and $E_e(\xi_e,\psi_e)$, we need to flip the rows ψ of *I* horizontally. Let $C_{x,y} \in \mathbb{R}^2$ be a CAD coordinate system in which *t* is to be vectorized. A captured pixel $E(\xi,\psi) \in I$ can be reconstructed as a point $\upsilon(x,y) \in C$ using the following relationship:

$$\upsilon(x,y) = E_\upsilon(\xi_\upsilon, \psi - \psi_\upsilon) \qquad (3.18)$$

From (3.18) we obtain the start point $s(x,y) = E_s(\xi_s, \psi - \psi_s)$ and end point $e(x,y) = E_e(\xi_e, \psi - \psi_e)$. Both pixels (points) are selected manually from an image of which the grid can consist of an enormous large number of pixel rows and columns up to the resolution of the image. Fact is, the higher resolution the input image consists of, the more accurate we can define the pixel of selection, see figure 3.2.5. On the contrary, the higher the resolution is, the smaller the pixel size becomes, and thus harder to manually

catch one pixel twice. The start point $s_{t_{i+1}}$ of an already captured module must absolutely overlap with the end point e_{t_i} of the module to be captured otherwise t_i and t_{i+1} will not form a continuous sequence which consequently causes problems with the 3D reconstruction of the captured 2D profile. More details about 3D reconstruction will be described in the final section 3.4.2.

Figure 3.2.5
Three images of a *c*-module of a basement in different resolutions. Left: 210 x 210 pixels, center: 42 x 42 pixels and the right image: 21 x 21 pixels.

To solve this problem we simply need to assign $s_{t_i} = e_{t_{i+1}}$ (we recall that it is a top-down directional capture). For the first module t_1 of a module sequence $S = (t_1,...,t_i,...,t_N)$ we need to capture both, its start- and end point whereas for the following t_i, for $i = 2,...,N$, we only need to capture the start points s_{t_i}.

Practically we do not require any further calculations to reconstruct the module sequence *S*. However, we want to display some informative properties of the module during capturing time such as the length l_t and the angle α_t. On the basis of E_s and E_e we can compute the length l_t using the *Euclidean distance*:

$$l_t = \left[\left(\psi_e - \psi_s \right)^2 + \left(\xi_e - \xi_s \right)^2 \right]^{\frac{1}{2}}. \qquad (3.19)$$

We know that $\psi_s > \psi_e$ which makes sure that $0 < \alpha_t < \pi$. The calculation of $\alpha = (E_e, E_s, h_0)$ with $h_0 = (\xi_s + 1, \psi_s)$ depends on the relation of ξ_s and ξ_e, and comprises of two cases:

1. $\alpha = \arctan\left(\frac{|\psi_s - \psi_e|}{|\xi_s - \xi_e|}\right)$ for $\xi_s \le \xi_e$ and (3.20)

2. $\alpha = \pi - \arctan\left(\frac{|\psi_s - \psi_e|}{|\xi_s - \xi_e|}\right)$ for $\xi_s > \xi_e$. (3.21)

After all module vectors are captured from the image and linked to each other as a sequence, we are ready to map the module profiles to each corresponding vectors. This process is identical to that presented in section 3.1.4.

The manual way of linking the modules independently from their types gives a certain combination flexibility which is necessary when dealing with reconstruction of temple elements with exceptional module structures. It is also aimed to apply for reconstructing special element structure of other temple styles. The left image of figure 3.2.6 for instance presents a basement of *Banteay Srey* temple with components $n_e = 2$ and $n_d = 1$, with exchanged functions such that *d* becomes the central module instead of *e* which is in Angkor Wat style absolutely uncommon.

Figure 3.2.6
Left: basement of Banteay Srey with special structure:
$S = (a,b,c,e,c,d,c,e,c,f,a)$.
Right: an eroded wall capital of a tower stage of Angkor Wat with a structure:
$S = (a,c,e,c,d,c,d,c,f,a,b,c,f,a)$.

Another purpose of this method is to deal with ATEs with uncommon module structure and eroded surface with the prerequisite that the start- and end point of each module as well as the module types are still identifiable. The right image of figure 3.2.6 shows an

example of this case. The software tool and how this method is implemented are described in subsection 4.3.2.

Both methods, the reconstruction based on the number of modules and the reconstruction using capturing modules from digital image, are based on predefined ideal modules. Therefore it is not to expect that the reconstructed model and the actual object absolutely match with each other. Nevertheless the feedbacks are more then reasonable and provide valuable initial information for further experiments and analysis. Reconstructing of temple elements which are in a good condition however, it is crucial to preserve their geometric information as much as possible. With this idea in mind we would like to propose a last reconstruction method which automatically detects the contour or element profile from the available digital image of the element.

3.3 Reconstruction Based on Image Segmentation

Image Processing, operations for changing, correcting, improving as well as analyzing an image, has been widely used for engineering and scientific purposes. Technologies and algorithms that have been found for implementation today are immense. In this section we will present an operation which belongs to *Image Segmentation*, a part of *Image Analysis*, for detection of a temple element profile from a digital image.

We will begin with some comments on the quality of an input image required for detection operation. An algorithm for detecting the temple element profile is then presented in section 3.3.2. We also introduce an algorithm for controlling the *levels of detail* of the element profile geometry at detection time in order to optimize the complexity of the final 3D model. Finally we show the conversion of the captured data into vector format in AutoCAD.

3.3.1 Quality of the Input Image

Segmentation operation is used to isolate particular objects from each other or from the image background. We implement an *amplitude segmentation* which is based on color components however, our purpose is not to produce a *binary image* nor to detect the whole closed boundary of the object of interest from an image as presented in [Pra07], [Jäh02], [Bax94]. We also will not discuss on image processing techniques to enhance the input image. Our goal is to identify the coordinates of each pixel element at the threshold between the temple element profile, which corresponds to a partial boundary of the whole object of interest, and its background (figure 3.3.1).

We have already seen in previous chapters that for all temple elements we have taken into account for discussion the module sequence are arranged upward forming the element profile. Both ends of the profile string define the row pixel section that contains information of the profile geometry. When we make a photography of the temple element to use as input image we obtain information of the element profile on the left- and right side of the object. This is mainly subjected to small scale elements such as column capital as shown in figure 3.3.1, column base, window grids, etc. We confine our discussion on segmenting element profile on the left hand side of the image.

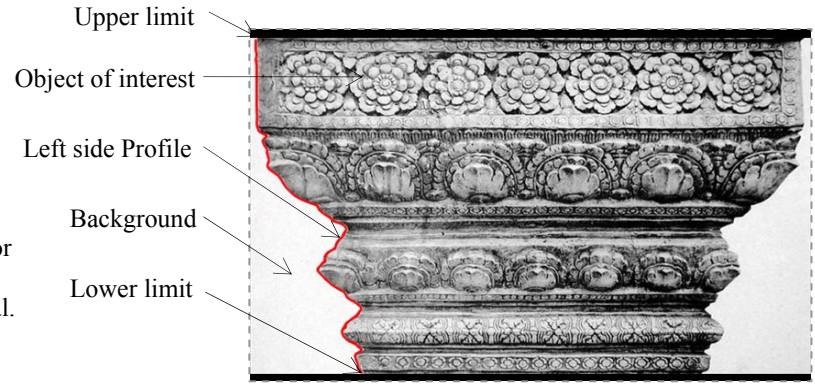


Figure 3.3.1
An input image sample for reconstruction of the profile of a column capital.

We define an input image as a grid $I_{\xi,\psi}$ with resolution of $\Xi \times \Psi$ pixels as in section 3.2. The pixel element of I is defined by $E(\xi,\psi,g)$ where $g \in Q$ denoting the gray value of E with $0 \le g \le 255$ and Q the finite number of the available gray level of

256 [Jän04]. A pixel that belongs to the background is denoted by E^B, and to the element profile by E^P. In order to segment the background from the element profile it is necessary that the gray values of the background pixels are different from that of the profile thus $g^B \neq g^P$. Because we are focusing only on capturing coordinates of the first pixel of the object at row ψ, it is not important whether $g^B < g^P$ or $g^B > g^P$. We discuss the case $g^B > g^P$ where the background is brighter then the object essentially at the threshold. Important is that $\xi^B \geq 0$ which means that for all rows $\psi = 0,1,...,\Psi - 1$ there must exist at least one pixel $E^B(0,\psi,g)$ to evaluate its gray value g^B with g^P of the following pixel $E^P(1,\psi,g)$.

3.3.2 Detection of Element Profile

The idea for detecting the threshold is very simple. For all rows $\psi = 0,...,\Psi - 1$ we capture two pixel elements $E_1(\xi,\psi,g)$ and $E_2(\xi+1,\psi,g)$, and evaluate the difference, of the gray levels $\Delta g = \left| g_\xi - g_{\xi+1} \right|$ of E_1 and E_2. The pixels of the background area appear uniform but in general Δg has non-zero value with thresholds of E_1^B and E_2^B remain minimal. This can be seen from the gray level graphs at three different parts of the image in figure 3.3.2. In order to tolerate these minimal gray value fluctuations within the background we define a limitation variable $\Gamma \in \mathbb{N}^{\geq 0}$ such that if $0 < \Delta g < \Gamma$ for all $\xi = 0,1,...,\Xi - 1$, then we define $E_2 \in E^B$. At a certain step we obtain the first enormous fluctuation of the gray values with $\Delta g > \Gamma$. In this case the threshold of the background and the object lies between E_1 and E_2. We then define:

$$E_\psi^P = E(\xi+1,\psi) \qquad (3.22)$$

as a pixel element that belongs to the object profile *P* at pixel column (abscissa) $\xi+1$, and row (ordinate) ψ, and skip the evaluation to the next row $\psi+1$, and repeat the same procedure.

When all pixel rows are evaluated, and all E^P are identified including their coordinates we are able to reconstruct the element profile by connecting the pixel (point) E^P_ψ with $E^P_{\psi+1}$ for all $\psi = 0,1,...,\Psi-1$. To vectorize the data packet in CAD system we must convert the coordinates from pixel coordinate system into CAD coordinate systems. Like the conversion in section 3.2.2, we let $C_{x,y} \in \mathbb{R}^2$ be the CAD coordinate system in which the vertex $\upsilon(x,y) \in C$ corresponding to the detected pixel $E^P(\xi,\psi) \in I$ is reconstructed. We use the following relationship:

$$\upsilon_\psi(x,y) = E^P_\psi(\xi, \Psi - \psi) \quad \text{for all } \psi = 0,1,...,\Psi-1 \qquad (3.23)$$

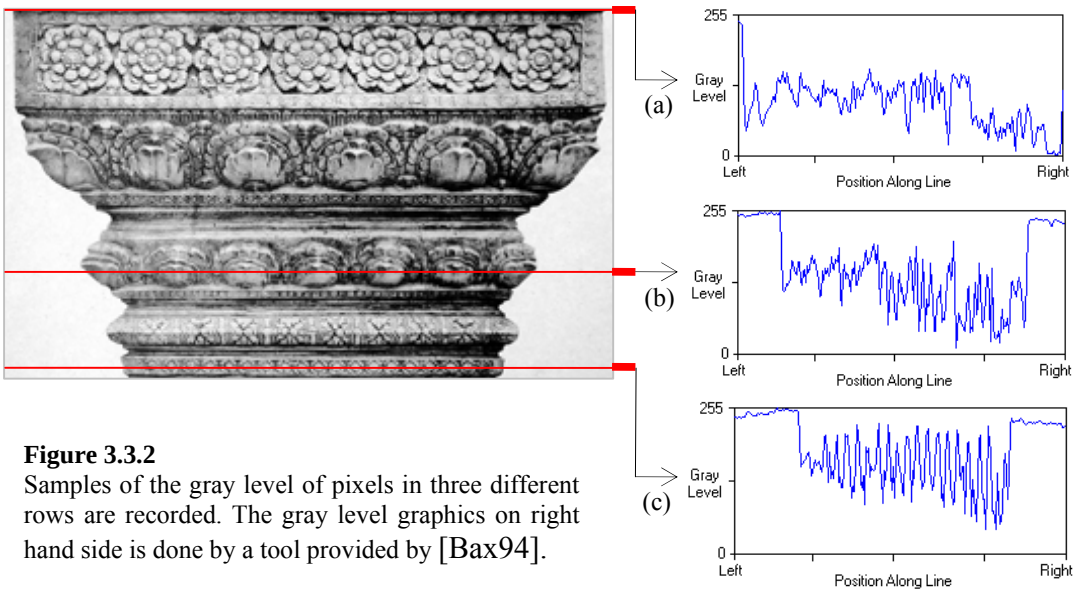


Figure 3.3.2
Samples of the gray level of pixels in three different rows are recorded. The gray level graphics on right hand side is done by a tool provided by [Bax94].

Input images taken from different direction or different location without artificial illumination provide different brightness and thus different gray level of the background. It is therefore necessary to redefine Γ to adapt to the background gray level of the actual input image by taking some samples of few pixels from the background. We will introduce a method to define Γ in section 4.4.

The algorithm for reconstruction of Angkor temple element profiles based on segmenting the object of interest presented above returns a highly detailed profile structure, and is proportional to the resolution of the input image. The number of vertices υ is equal to the number of rows that image *I* consists of. An input image with a standard resolution of 800×800 pixels would produce 800 vertices and 799 line segments for a single profile. 2D reconstruction with such details is still acceptable but it would quickly lead to data abundance at the 3D level.

On the other hand, some part of a module profile, e.g. the middle segment of *a*-module profile is represented by a single straight line. The arcs of *b*- and *f*-module for instance could also be represented by few line segments, and do not require unnecessary numerous lines segments to form them.

In the following section we present two algorithms for controlling the levels of detail of the element profile including their advantages and disadvantages.

3.3.3 Optimizing Geometry and Controlling Levels of Detail

The first algorithm is aimed to optimize the geometry of the element profile. The control for optimization is based on evaluating the angle difference of two adjacent line segments. For the case where the angles difference is minimal, we consider both lines as one single line. In this case the second line segment is ignored, and the geometry of the element profile becomes one edge less complex. Otherwise it is defined as part of the element profile, and thus must be preserved. Thus we need to define a deviation tolerance of the angles in order to control the accuracy of the reconstruction, and the levels of detail of the object geometry.

Let $E_{\psi}, E_{\psi+1}$ and $E_{\psi+2}$ be three pixel elements representing the nodes of the two adjacent line segments $l_{\psi} = (E_{\psi}, E_{\psi+1})$ and $l_{\psi+1} = (E_{\psi+1}, E_{\psi+2})$. l_{ψ} and $l_{\psi+1}$ form with horizontal lines $\left[E_{\psi+1}, \left(\xi_{\psi+1}+1, \psi+1\right)\right]$ and $\left[E_{\psi+2}, \left(\xi_{\psi+2}+1, \psi+2\right)\right]$ the angles γ_{ψ} and

$\gamma_{\psi+1}$ respectively. We define the difference of γ_{ψ} and $\gamma_{\psi+1}$ by $\Delta\gamma = \left|\gamma_{\psi} - \gamma_{\psi+1}\right|$. The deviation tolerance variable is denoted by $T \in \mathbb{N}$, where $0 \le T \le \frac{\pi}{2}$.

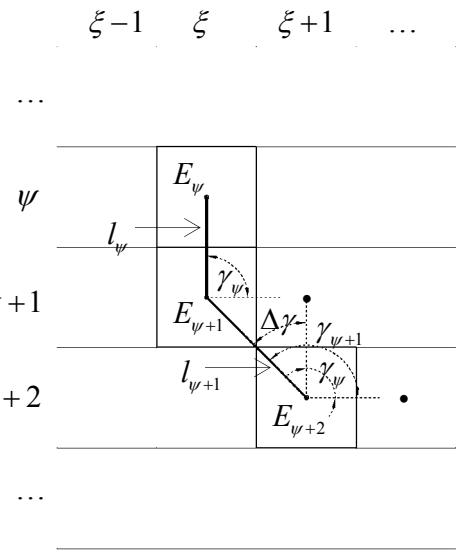


Figure 3.3.3
Representation of three detected pixels $E_{\psi}, E_{\psi+1}$ and $E_{\psi+2}$ from image $I_{\xi,\psi}$, and their angles γ_{ψ}, $\gamma_{\psi+1}$ and $\Delta\gamma$.

Each evaluation interval requires those three detected pixel elements as input data. As we have mentioned above, the evaluation returns only two type of results:

1. $\Delta\gamma > T$: $l_{\psi+1}$ deviates too much from l_{ψ}, and they must be considered as two different line segments. In this case the coordinates of all three pixels are saved in the temple element profile database, and $E_{\psi+2}$ is set as the initial pixel for evaluating the following line segments $l_{\psi+2}$ and $l_{\psi+3}$.

2. $\Delta\gamma < T$: l_{ψ} and $l_{\psi+1}$ deviates minimal from each other. Both lines will be unified as one line in such a way that $E_{\psi+1}$ will be rejected and removed from the database. The two remained points together form a new line segment $l'_{\psi} = (E_{\psi}, E_{\psi+2})$ which will be saved in a so-called *waiting list*, and is ready for the next evaluation. As soon as the system detects the next valid pixel, it is again added to the list to do the next evaluation, and so forth.

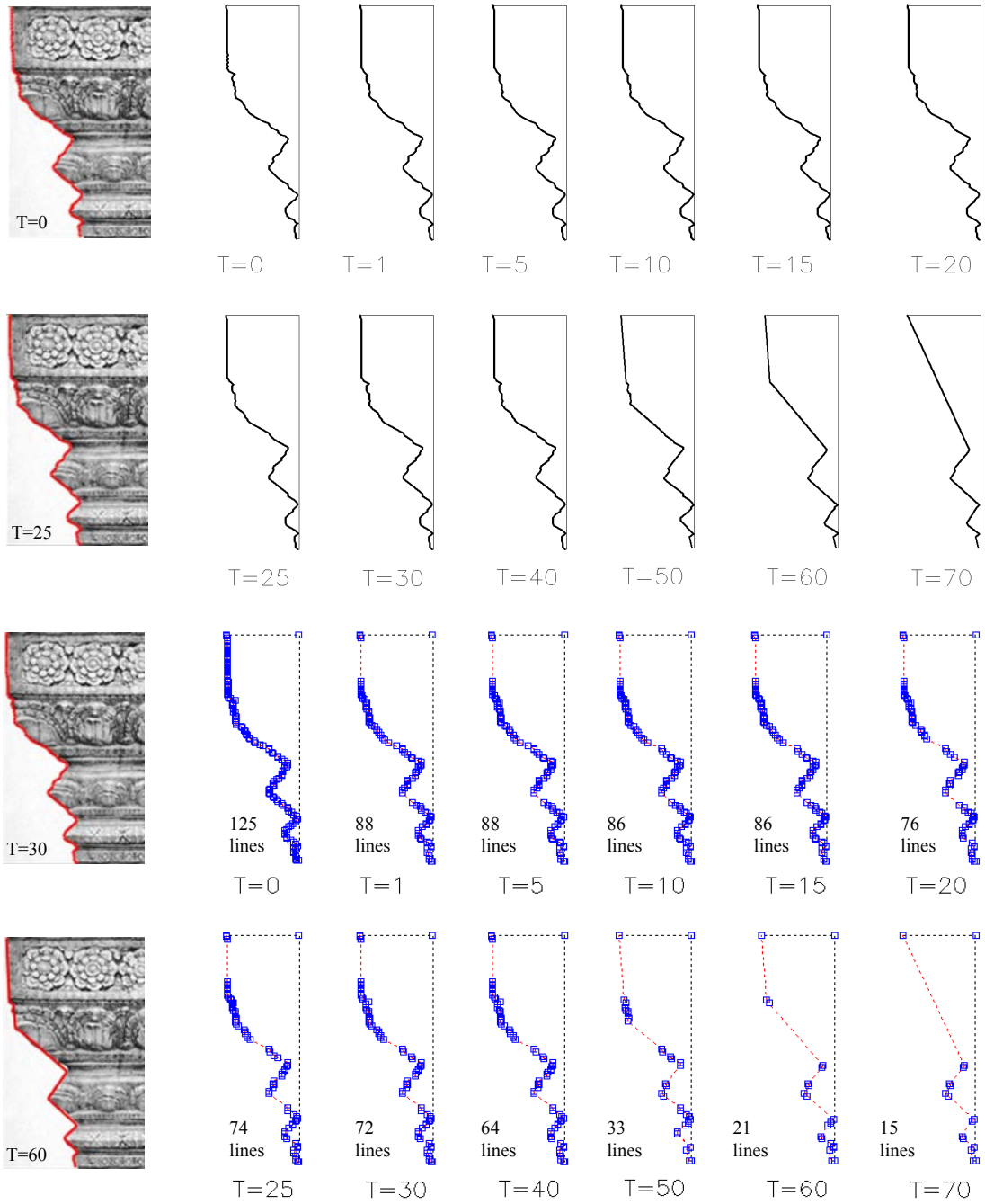


Figure 3.3.4
12 samples for detection of a column capital profile from an input image with a resolution of $I_{\xi,\psi} = \Xi \times \Psi = 300 \times 186$. The difference of the complexity can be seen from the number of lines identified by vertices shown as small squares along the profiles.

Figure 3.3.4 shows 12 different simplification levels controlled by 12 different value of *T*, produced by the software tool which we are going to discuss in detail in chapter 4. In this example we measure the angles in degrees in order to ease observation of each simplification steps at small changes of *T* value. We see that solely with one degree limitation *T*=1 we already can reduce the complexity by 37 lines. We also can observe that vertices lying within one straight line are removed, and are only necessary at corners and curves. This comes out clearly in our examples when $T \geq 20$. The optimal simplification level amongst all 12 is *T*=40. For $T > 40$ too much geometric information get lost although the profile still consists of 64 line segments.

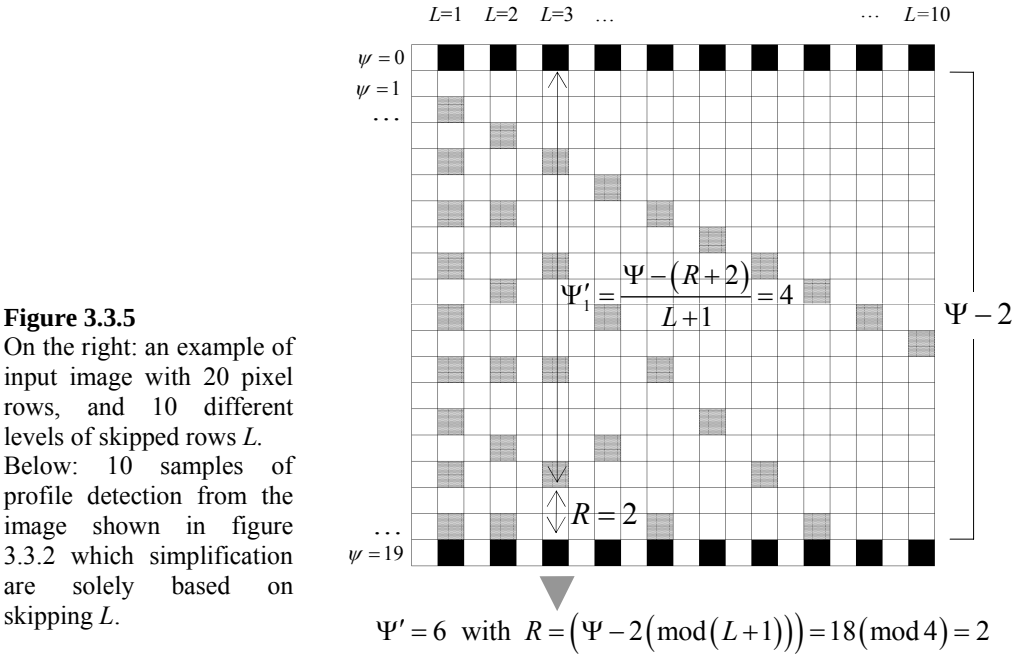
In this simplification method all pixel rows are examined, and thus returns a considerable precise feedback. Even at a level of *T*=60 the system preserve the characteristic of the original temple element profile with all modules (primitive) shape revealed. For a certain purpose, such as virtual realistic representation of very complex 3D structure with texture mapping, this level would be the most reasonable one.

The second simplification algorithm is based on linear vertex reduction independently from the gray level of each pixel. Here we skip a certain number of pixel rows at each pixel detection interval. Doing like this we can reduce the calculation time immensely, and is proportional to the number pixel rows that have been skipped. This method is very useful especially when we deal with high-resolution input images to examine different levels of detail from higher performance downward.

Let $L \in \mathbb{N}^{\geq 0}$ be the number of pixel rows to be skipped. In addition we define $\Psi', R \in \mathbb{N}^{\geq 0}$ as the new number of pixel rows of image *I* at *L* skip and the reminder respectively. It is crucial that the number of pixel rows to be evaluated is computed in advanced in order to define the limit of the skipping process. Ψ' is then calculate as following:

$$\left.\begin{aligned} \Psi' &= \frac{(\Psi - 2) - R}{L + 1} + 2 \\ \text{with reminder } R &= (\Psi - 2)\left(\bmod(L + 1)\right) \end{aligned}\right\}. \qquad (3.24)$$

Figure 3.3.5 presents an example of an input image $I_{\xi,19}$ with 20 pixel rows, and shows the homogenous part as well as the corresponding reminder with different levels $L = 1,...,10$ in different pixel columns. The black colored pixels define the first and the last nodes of the structure and the gray colored are the evaluated pixels.



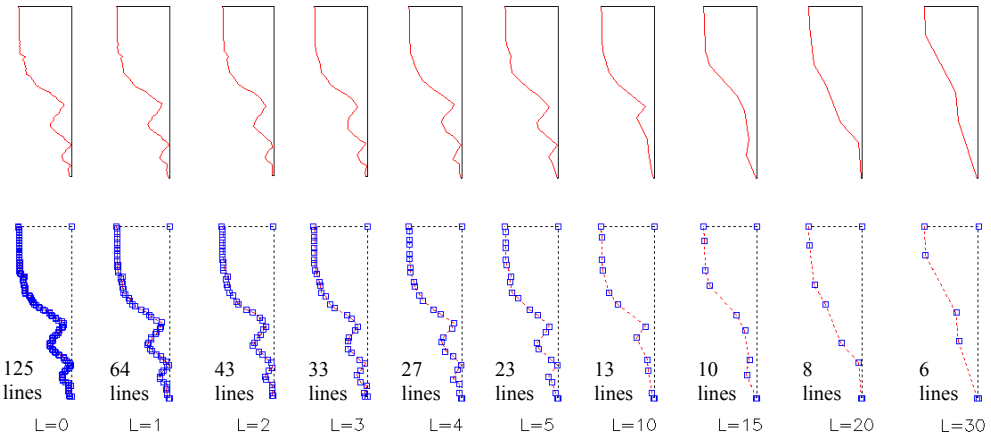


Figure 3.3.5
On the right: an example of input image with 20 pixel rows, and 10 different levels of skipped rows *L*. Below: 10 samples of profile detection from the image shown in figure 3.3.2 which simplification are solely based on skipping *L*.

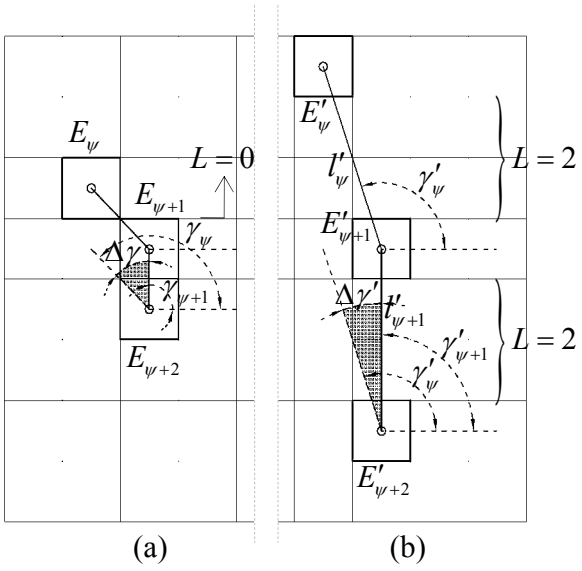
Our experiments show that this algorithm together with the previous one can enhance the effectiveness of the simplification. By increasing the scalar ψ (when skipping pixel rows) whereas scalar ξ remains unchanged $\Delta\gamma$ gets smaller and thus less "sensible" in the difference $\left|\xi_{E_{\psi+1}} - \xi_{E_\psi}\right|$. Let us denote $E'_\psi, E'_{\psi+1}$ and $E'_{\psi+2}$ the new three pixels at $L \neq 0$. From these pixels we obtain two lines l'_ψ and $l'_{\psi+1}$ which form with a horizontal line the angles γ'_ψ and $\gamma'_{\psi+1}$, and further the difference of both angles $\Delta\gamma'$ (see Figure 3.3.6 for the case of L=2). Based on Pythagoras theorem we get:

$$\cot(\Delta\gamma) = \frac{\left|\psi_{E_{\psi+1}} - \psi_{E_\psi}\right|}{\left|\xi_{E_{\psi+1}} - \xi_{E_\psi}\right|} \text{ and } \cot(\Delta\gamma') = \frac{\left|\psi_{E'_{\psi+1}} - \psi_{E'_\psi}\right|}{\left|\xi_{E'_{\psi+1}} - \xi_{E'_\psi}\right|} \qquad (3.25)$$

since $\left|\xi_{E'_{\psi+1}} - \xi_{E'_\psi}\right| = \left|\xi_{E_{\psi+1}} - \xi_{E_\psi}\right|$ and $\left|\psi_{E'_{\psi+1}} - \psi_{E'_\psi}\right| > \left|\psi_{E_{\psi+1}} - \psi_{E_\psi}\right|$ we obtain $\Delta\gamma' > \Delta\gamma$.

Figure 3.3.6
Comparison of angle difference $\Delta\gamma$ and $\Delta\gamma'$ of two groups of two lines, each consisting of three pixels $E_\psi, E_{\psi+1}, E_{\psi+2}$ and $E'_\psi, E'_{\psi+1}, E'_{\psi+2}$ with :
$\left|\psi_{E'_{\psi+1}} - \psi_{E'_\psi}\right| = \left|\psi_{E_{\psi+1}} - \psi_{E_\psi}\right| + 2$.

$\Delta\gamma' < \Delta\gamma$



The result of implementing both simplification methods together at the same time is presented in figure 3.3.7. In figure 3.3.8 we compared all three methods by selecting three levels of *T* and *L* from the already obtained results. We observe in the highest level of detail for instance; algorithm 1 produces 33 lines, algorithm 2 produces 27 lines and combined algorithm only 23 lines but the best profile structure.

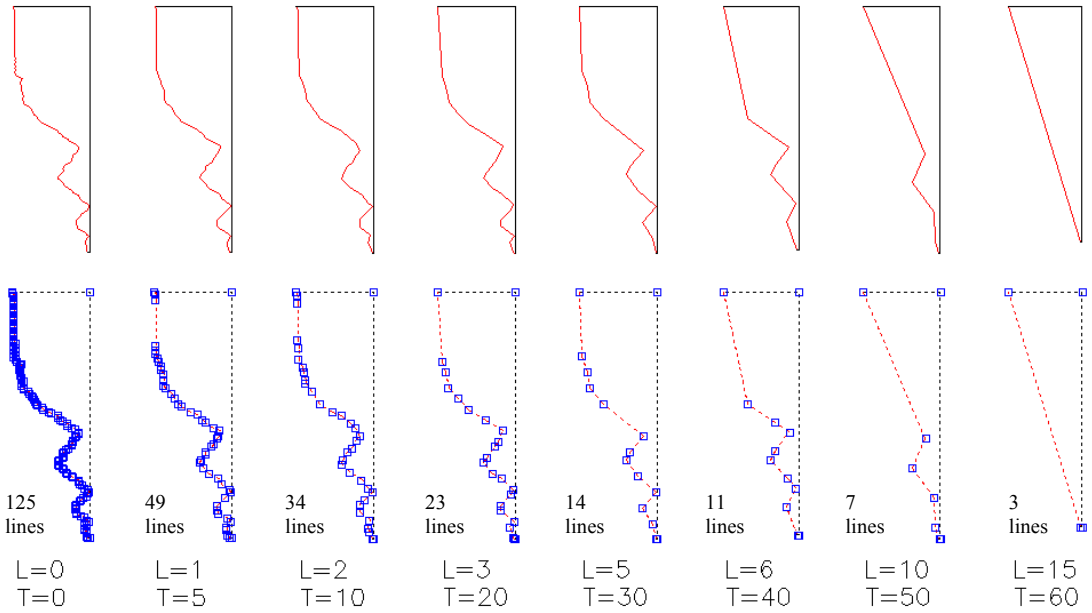


Figure 3.3.7
Eight samples of simplification using both method, evaluating T and skip pixel rows by L at the same time.

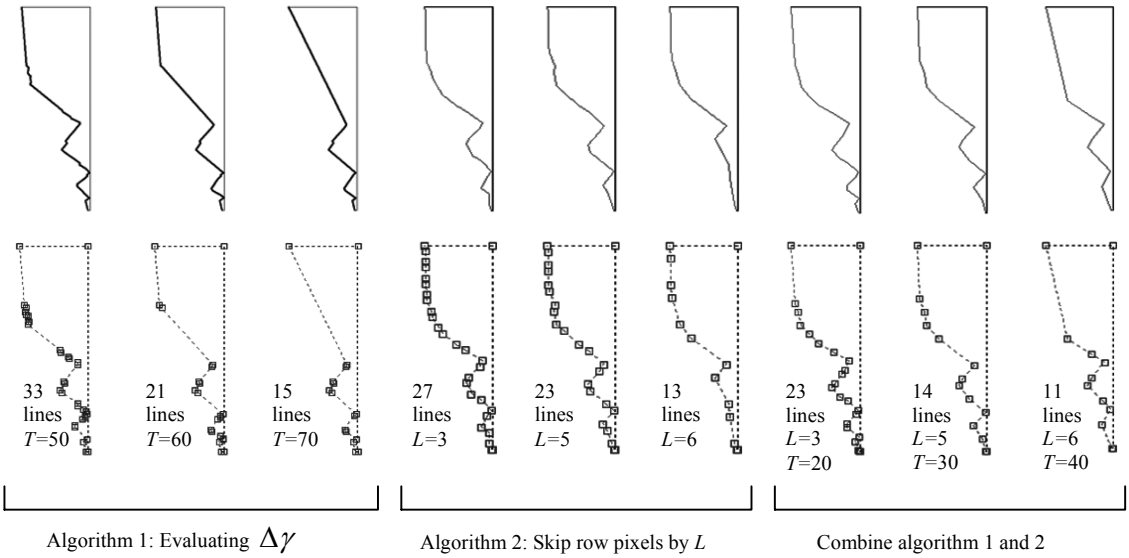


Figure 3.3.8
Comparison of the three simplification algorithms; evaluating $\Delta\gamma$ (first three profiles), skipping pixel rows by L (three middle profile), and a combined of both algorithms (last three profiles), each at three different level settings.

3.4 3D Reconstruction of Temple Elements

In this last section of chapter 3 we will discuss the 3D reconstruction of temple elements using the 2D profiles we obtained in section 3.1, section 3.2 and 3.3. A *Constructive Solid Geometry* approach is partially applied for 3D modeling of ATE, and we provide an introduction in section 3.4.1. Section 3.4.2 we present an approach to make the *opened* line segments of the element profile become a *bounded* region. Then we discuss on 3D operations including the so-called *extrusion* operation that performs the 3D generation of solid bodies from 2D cross sections, and is mainly applied for 3D reconstruction of Angkor temple elements. Finally we do the conclusion in section 3.4.4.

3.4.1 Constructive Solid Geometry (CSG) of ATE

3D reconstruction of ideal Angkor temple elements is suggested to implement the *building block* approach where each block is represented in solid geometry. This method allows us to preserve complete geometric information of the objects in order to flexibly modify the objects using 3D operations such as *cut or slice, union, subtraction, intersection* and other Boolean operations during modeling time. Thus each building must be a *valid solid* reconstructed from a bounded set. Building blocks of an element are separately modeled based on distinct simple but sufficient bounded point sets which then can be generally combination together as a whole. Such model is called *Constructive Solid Geometry* [Män88].

The *CSG tree* representing a single Angkor temple element generally has two leaves (primitives): leave (a) the *sweeping* of a bounded element profile Π of P (see details in next section) along the ground floor projection (*extrude path*) Q of the element, leave (b) sweeping of a bounded Q by the height h of P. The result (c) is the combination (union) of (a) and (b), see figure 3.4.1. The CSG tree can be represented as following:

$$< basement\ tree > \ ::= \ < solid\ of\ basement\ profile > \ < union > |$$
$$< solid\ of\ swept\ of\ basement\ ground\ floor\ plan\ offset >$$

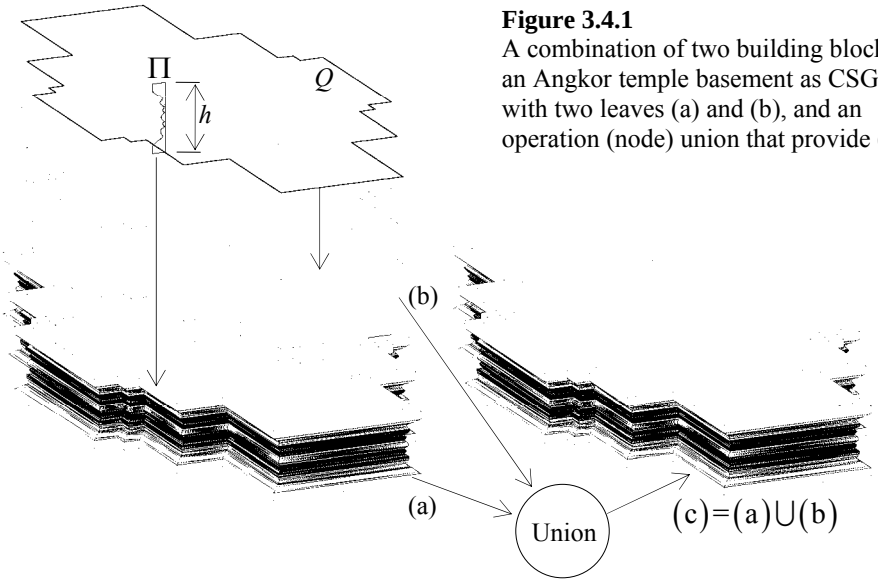


Figure 3.4.1
A combination of two building blocks of an Angkor temple basement as CSG tree with two leaves (a) and (b), and an operation (node) union that provide (c).

Results we obtained in section 3.1, 3.2 as shown in figure 3.1.17, and section 3.3 shown in figure 3.35 (the red color profile) are non-closed 2D line segments. In the following section we discuss on converting the element profile into a bounded 2D point set as well as abstract representation of 3D reconstruction of ATE profile and the filling volume as shown (a) and (b) respectively in figure 3.4.1.

3.4.2 Reconstruction of Bounded Element Profile

Before an element profile can be used to reconstruct a 3D solid object the profile must be modified into a bounded (and *closed*) *region* or *limina* [Män88]. To do this we propose to add line segments (polyline) such that the start point and end point of the profile are individually joined with both ends of the line segments. This can be done differently however, it is important that both objects forms one single bounded region. Figure 3.4.2 (b) for instance gives an example of an invalid combination with a single line that results three different regions $R1$, $R2$ and $R3$. In contrast to (c) and (d), profiles P are joined with two and three line segments respectively, and each form valid regions. Sample (c) is

compared with (d) a better solution in terms of virtual realistic representation using basic *plane* or *cubic* texture mapping (a single image of the profile façade) to the final 3D solid of the element because the offset distance $o = 0$, and thus the warped texture is not visible. We will give an example on this subject in chapter 4. For 3D reconstruction of temple element in this research we focus on sample (d): a combination of profile *P* with three line segments.

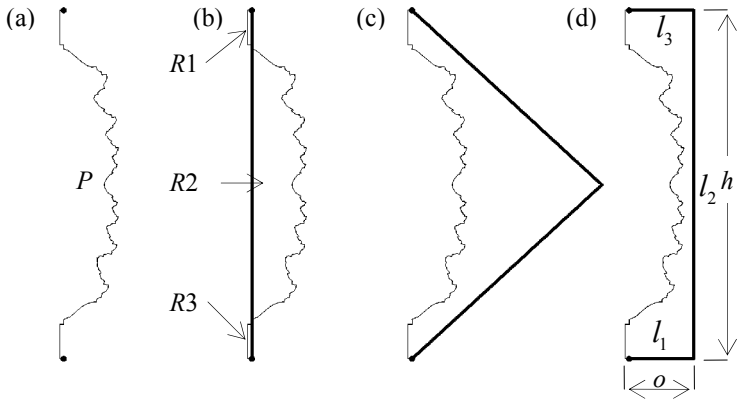


Figure 3.4.2
(a) unbounded basement profile. (b), (c) and (d) are bounded by closing the line segments with one, two and three lines segments respectively.

In order to model a valid bounded region of profile *P* for 3D generation we should discuss on some properties of the additional line segments in advanced. There are two parameters we should consider: the height *h* (height of *P*) and the offset distance *o* from the start point of *P* (see figure 3.4.2 (d)). The height *h* is defined as the length of the middle line segment and can be calculated using definition 2.3.7. Offset distance *o* must be defined in respect to property of the *extrude path* or the ground floor plan of the corresponding element which *P* is going to sweep along.

Figure 3.4.3 gives an example on problematic of the offset distance *o*. In this figure we show seven different samples of sweeping a profile, as a simple rectangle each with same height *h* but a varying *o* parameter, along a given ground floor plan of a basement. Both bolded lines, the first outer most and second the offset of the outer most boundary by *o*, represent the boundaries of the swept profile; the 3D solid.

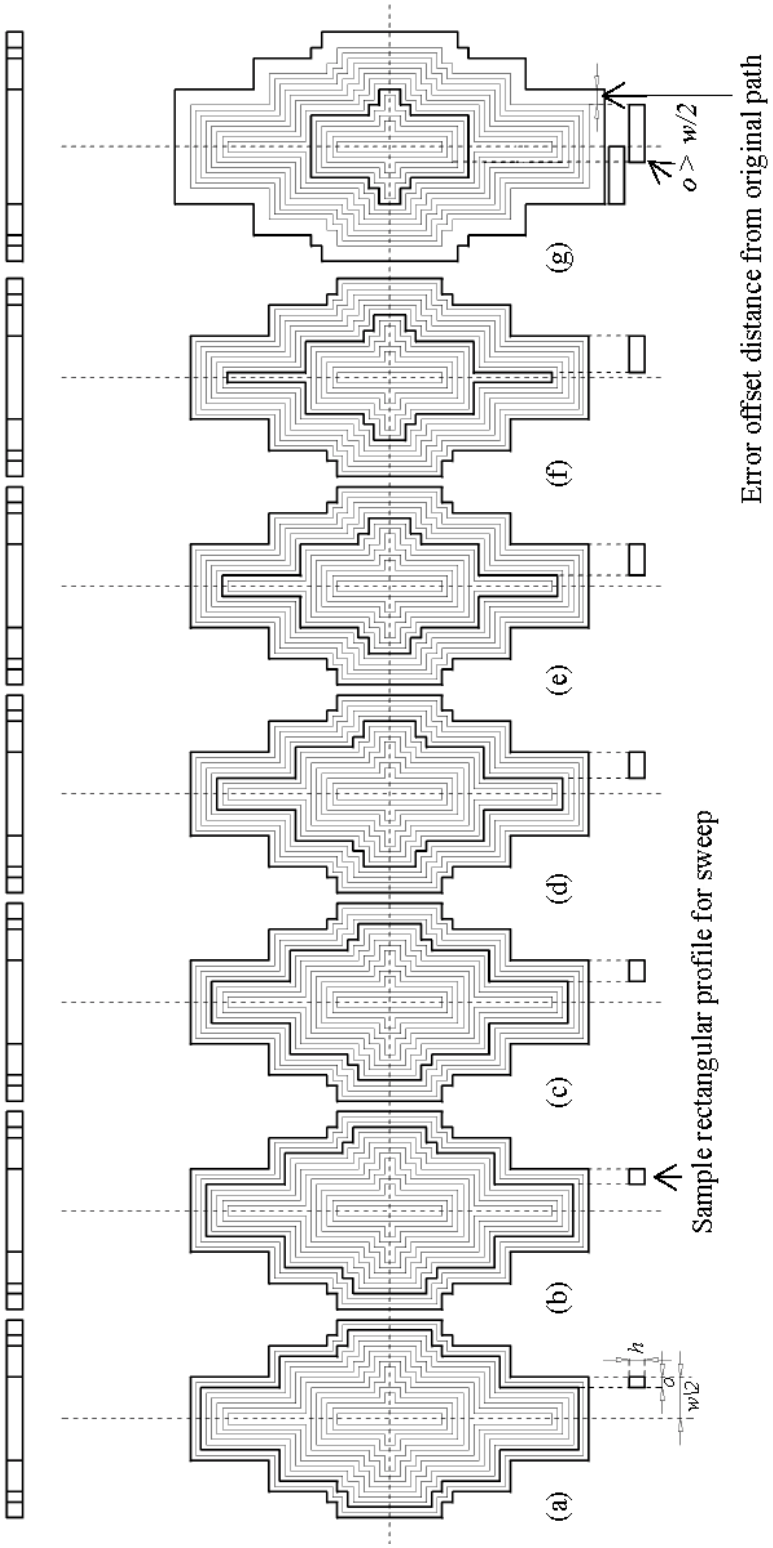


Figure 3.4.3
On top are front views of the below bold marked border seven of solid samples of sweeping rectangular profiles of height h, offset distance o and sweeping along same ground floor plan (path) of a basement. We see case (g) produce error because the offset distance $o > w/2$.

The relationship of o with the ground floor plan can be observed from the inner boundary of the solid. We increase the value of o at each step in such a way that the inner boundary of the solid overlaps one of the offset lines (offset by a constant value of $2 \times o$ to the inside direction of the ground floor plan). We observe that the outer border of the 3D solid correctly remains fit with the ground floor plan until step (f). In contrast to case (g) where $o > w/2$ (w is the width of the narrowest part of the ground floor plan; in the example of figure 3.4.3 are the upper- and lower-most parts of the ground floor plan) the modeler of the system (in this case AutoCAD modeler) produces an error 3D model that is larger in width and length than the given ground floor plan. Thus we define the maximum offset $o_{\max} < w(Q)/2$ where $o_{\max} \in \mathbb{R}^{>0}$, and w denotes the narrowest width of a ground floor plan Q which profile P is swept along (see figure 3.4.3).

On the other hand we need to discuss on the allowed minimum of $o = o_{\min} \in \mathbb{R}^{>0}$ that still preserve the profile boundary as a single bounded region. We recall from section 2.2.3 that module of type c is the only one concave module of which geometry reaches far deep inside the volume of the element. Another considerable property of module c in this context is that c plays a roll as connector of almost all convex modules, and thus for any geometric type of Angkor temple element profile, the geometry of one c_i always lies at the right most position compared to the other types. The index i varies by type of temple element. Figure 3.4.4 gives some example of our description.

We propose for this reason to find out the right most lying c module type in the sequence P by comparing the abscissa value of its start point $x_{s^{c_i}}$. The minimum offset $o_{\min}$ is defined by:

$$o_{\min} = x_{s^{c_i}} + l_{c_i} < o_{\max} \tag{3.26}$$

where t_{c_i} is the length of c_i defines (see figure 3.4.4).

The three line segments l_1, l_2 and l_3 now have the following general parameters:

$l_1 = l(s_{l_1}, e_{l_1})$ with $s_{l_1} = (x_{s^{t_1}}, y_{s^{t_1}})$, $e_{l_1} = (x_{s^{c_i}} + l_{c_i}, y_{s^{t_1}})$,

$l_2 = l(s_{l_2}, e_{l_2})$ with $s_{l_2} = (x_{s^{t_1}} + l_{c_i}, y_{s^{t_1}})$, $e_{l_2} = (x_{s^{c_i}} + l_{c_i}, y_{e^{t_N}})$ and

$l_3 = l(s_{l_3}, e_{l_3})$ with $s_{l_3} = (x_{s^{c_i}} + l_{c_i}, y_{e^{tN}})$, $e_{l_3} = (x_{e^{tN}}, y_{e^{tN}})$.

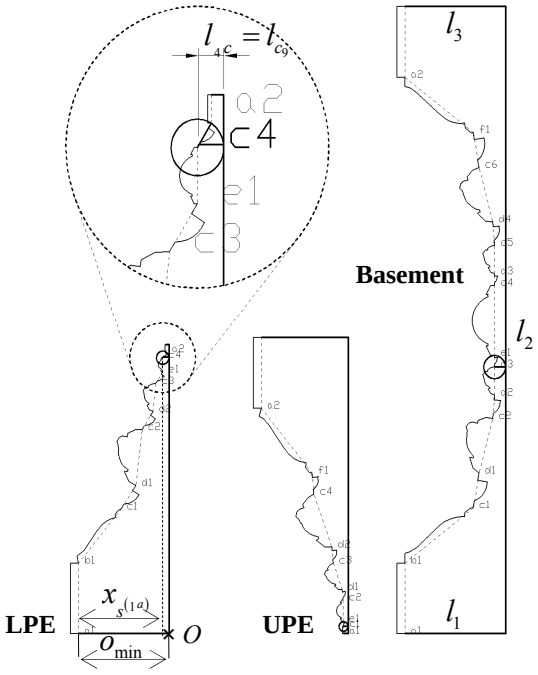


Figure 3.4.4
Lowest limit of $o = o_{\min}$ in relation to the abscissa $x(s_{c_i})$ of the start point of module vector c_i at *i*-th index.

For the reason that reconstruction algorithm presented in section 3.2 is also based on module and produce the same result as that from section 3.1, this method is also define for modeling a bounded region with that algorithm as well.

Differently must be handled with the profile obtained from image segmentation because this reconstruction method is not based on modules but sequence of line segments. Instead of identifying the identity of the right most lying line segment we simply "scan" for the largest value of abscissa ξ_i of detected point, and define $x_{\max} = \xi_i$, for $i = 0,...,n-1$ where n defines the total number of pixel rows that have been evaluated from the input image.

The algorithm for determine $x_{\max}$ has the following structure:

Algorithm 3.4.1 (determine abscissa $x_{\max}$ of the right most captured pixel)

1. Set initial value of $x_{\max} = 0$

2. For all column index ξ_i for $i = 0,...,n-2$ of the detected pixels E_ψ^P see equation (3.22), do the following:

- If $x_{\max} \neq 0$ and $i \leq n-2$, evaluate the pair ξ_i and ξ_{i+1} and select following case:
- Case $\xi_i > \xi_{i+1}$: set $x_{\max} = \xi_i$
- Case $\xi_i < \xi_{i+1}$: set $x_{\max} = \xi_{i+1}$

The minimum offset distance $o_{\min}$ then can be define by adding one unit (length of one pixel) to $x_{\max}$ thus $o_{\min} = x_{\max} + 1$. The three line segments can then be modeled using the same method that has been applied for reconstruction of bounded region of module based element profile as shown above.

3.4.3 3D Reconstruction of ATE

The 2D profile *P* has become a bounded (and closed) region, and we can reconstruction a 3D solid model of the temple element profile by spanning the 2D profile along a given ground floor plan projection of the element. Let Π be the point set of the closed polygon of profile *P* in a two dimensional Euclidian space: $\Pi \subset \mathrm{E}^2$, and let us denote the ground floor plan *Q* as a sequence of *n* line segments v_i for $i = 0,1,...,n$. Additionally we define a point $O \in \Pi, v_i$ as the origin to support Π on vector v_i.

The operation of extrusion a polygon along a given path that is implemented by AutoCAD modeler differs from sweeping operation (*Minkowski sum with interval*) presented by [Pao03] in a way that the polyhedron to be translated do not change direction. In our case the plane of Π that is to be swept along vector v_i must be perpendicular to v_i thus $v_i \perp \mathrm{E}_i$. We consider extrusion of Π within two adjacent interval v_i and v_{i+1}, where the outside angle $\measuredangle(v_i, v_{i+1}) > \pi$ and $x_O \geq x_\Pi$ do not cover the space at the corner of both vectors, see figure 3.4.5 (a) and (b). In order to fulfill the

incomplete volume we must calculate the *intersection* volume $I_{i,i+1}$ of the extended volume V_i and V_{i+1}, see (c) and (d).

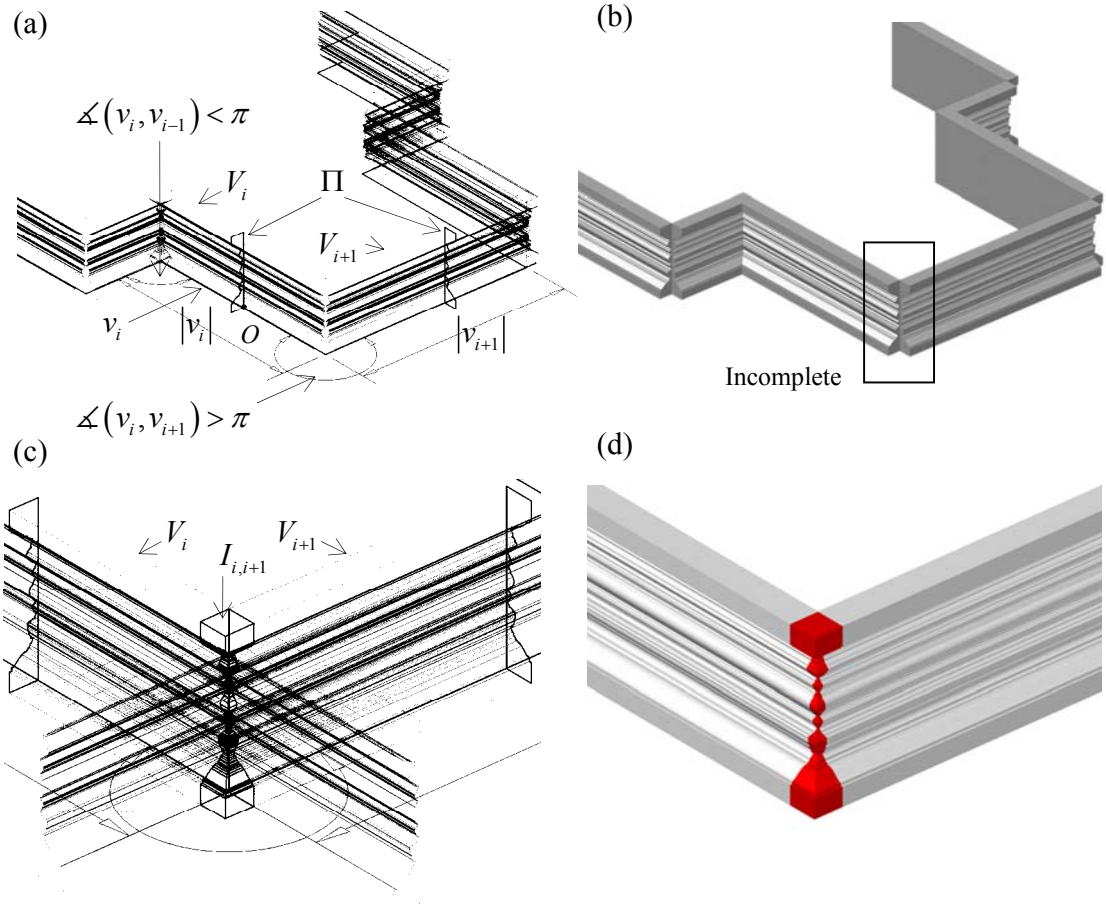


Figure 3.4.5
(a) Sample for sweeping Π with interval v_i and v_{i+1}, resulting two solid volume V_i and V_{i+1} respectively. (b) Representation of uncompleted volume at a corner. (c) Prolonged solid of V_i and V_{i+1} to find out their intersection which is shown in shaded format in (d).

Before we present the abstract model of the complete extruded 3D model of a Π along a ground floor plan Q with n intervals (line segments), let us keep in mind the following proportion.

Proportion (Sweep): Based on [Pao03] consider the interval $[v_i]$ of a ground floor plan projection of a temple element as two segments $[-v_i/2, v_i/2]$ with a midpoint *O* be the origin of the point set $\Pi \in \mathrm{E}_i^2$ of an element profile to be spanned by v_i with $O \in v_i, \Pi$ and $v_i \perp \mathrm{E}_i$. We obtain V_i by sweep operation:

$$Sweep(v_i) : \mathrm{E}_i^2 \to \mathrm{E}^2 : \Pi \mapsto \Pi + [v_i] \quad \text{where}$$

$$V_i = \Pi + [v_i] = \left(Sweep(-v_i/2) \circ Sweep(v_i/2)\right)(\Pi). \qquad (3.27)$$

Intersection of sweeping: Two solids V_i and V_{i+1} intersect each other in a subset $I_{i,i+1}$ is called the *logical intersection* and is described as: (figure 3.4.5 (c))

$$I_{i,i+1} = V_i \bigcap V_{i+1} \qquad (3.28)$$

The 3D solid *V* (object (a) in figure 3.4.1) of sweeping a point set Π along *Q* as a union of a pair wise union of V_i and $I_{i,i+1}$ we have obtained in (3.27) and (3.28) respectively for $i = 1, ..., n$ and $I_{n,n+1} = I_{n,1}$:

$$V = \sum\nolimits_n V_i \bigcup I_{i,i+1} \qquad (3.29)$$

Finally the 3D solid of the whole temple element need to be completed by reconstructing a filling volume *W* (object (b) in figure 3.1.4), and unify it with *V*. In some cases such as that of reconstructing a column or wall, volume *W* has more advanced form and function if compared to the basement. For a column for instance *W* of a column base is used to fulfill the inner volume however, its extrusion by the height of the column body and column capital will provide a complete column as shown in figure 3.4.6 (b) and (c). In this figure we also see that the inner volume of a wall base and wall capital must remain open. The solid objects of the wall base and wall capital (extruded profile) are simply glued to the wall body, and thus do not require additional *W* solid to fill in the bounded solid.

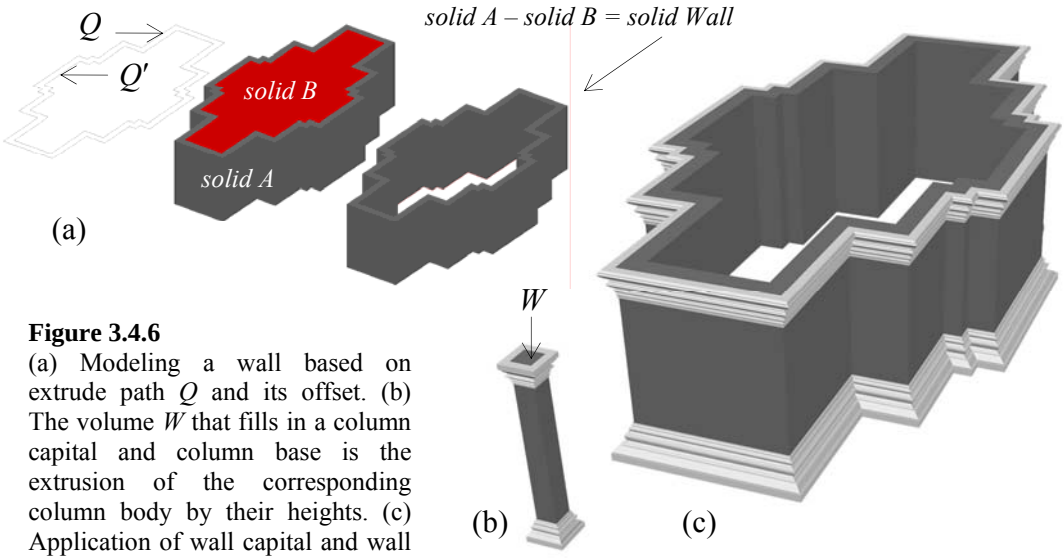


Figure 3.4.6
(a) Modeling a wall based on extrude path Q and its offset. (b) The volume W that fills in a column capital and column base is the extrusion of the corresponding column body by their heights. (c) Application of wall capital and wall base with wall body.

Although different types of ATE require different treatments to complete its 3D geometric property, it is crucial to notice that the operations for their production have much in common. All of them are based on the extrude path Q of the profile point set Π. The basement and column for example can be completed by directly extruding the closed extrude path Q that was used to generate V. LPE and UPE mainly wall base and wall capital on the other hand is to completed with a wall which either can be obtained by the *subtraction* of two solid objects, say *solid Wall* = *solid A* – *solid B* where *solid A* is the extrusion of Q by the height of the *solid Wall*, and *solid B* is the extrusion of an offset Q' of Q by the thickness of the wall then extrude by the same height of *solid Wall* with area $A(Q) > A(Q')$, see figure 3.4.6 (a). Another alternative is to extrude a give cross section of the wall alone Q.

3.4.4 Conclusion

In this chapter we have presented three different algorithms for reconstruction of 2D Angkor temple element profile, and followed by 3D reconstruction using the obtained 2D profile. The first two are modules based combinatorial algorithms from which the first

automatically reconstruct ATE profile by analyzing the given type and number of modules. This algorithm implements predefined construction concepts and rules that the ancient Khmer architects and artists once might had in mind. The second allows the user to manually pick each module from available image of the element, and it is thus an algorithm of a combined concept of reconstructing ideal temple element using input data from existing element. Last algorithm does not rely on the predefined ideal modules and construction rules. It detects the element profile from the available input image of a temple element. This algorithm is the best for preserving and depicting geometric information of existing element in the 2D vectorized profile, and further in the 3D model.

We finally introduced some suggested methods and operations for converting the obtained 2D element profile into a final 3D solid object in respect to the structure of Khmer temple and available 3D CAD operations.

The following chapter we are going to present our software tool which was designed for implementation of the algorithms and methods we have introduced in this chapter.

Chapter 4

Angkor Temple Generator Software Tool

In this chapter we discuss all about our self-written *Angkor Temple Generator (ATG)* software tool. We will start with the section 4.1 in which rough description on the architecture of ATG and how it is designed in *Visual Basic for Application (VBA)* as well as the interaction between ATG and its parent tool AutoCAD. Here we give some instructions on how to access ATG codes for further potential improvements. The library of ATG, its pipeline structure, the user interface and other features of ATG are shown in this section too. The following three sections; 4.2, 4.3 and 4.4 we demonstrate the three methods for reconstruction of the ATE profiles we have introduced in chapter 3, and will discuss their practical advantages and drawbacks. Section 4.5 we show how to generate a 3D model of the element and introduce methods on how ATG can be best implemented. The last section 4.6 presents some suggestions on improvement of the ATG tool.

4.1 Architecture of ATG

Autodesk provides two *Integrated Development Environment (IDE)* in order to customize and extend controls over AutoCAD; Visual Basic for Applications (VBA) and *Visual LISP*. In section 4.1.1 we will give an overview of VBA, and point out the reason we choose VBA for our project. and describe the functions of *Macros* as components of ATG in section 4.1.2. The graphics interface of the tool is shown afterwards in section

4.1.3 with special features that make ATG powerful and user-friendly. The final section shows how to load ATG project and how to execute the application. Before discussing about VBA, we would like to point out the advantages of building ATG on an existing host CAD tool instead of writing a standalone application.

The major inspiring factor is that we want to bring out our software to a professional level by providing mutual exchange of significant features of both applications. Powerful CAD tool equips with a wide variety of features such as Boolean operations, precise controls on moving, placing, rotating, mirroring, etc. as well as absolutely precise controls on geometric measurement of 2D and 3D objects to be modeled. Dealing with highly complex structured models requires an ingenious system of displaying the monitoring objects in order not to get lost in "the jungle of wires" when displaying as *Wireframe* view format at modeling time (see figure 4.1.1). All these available CAD features are based on highly complex mathematical calculations and programming tasks which for this project does not make any sense to reproduce again.

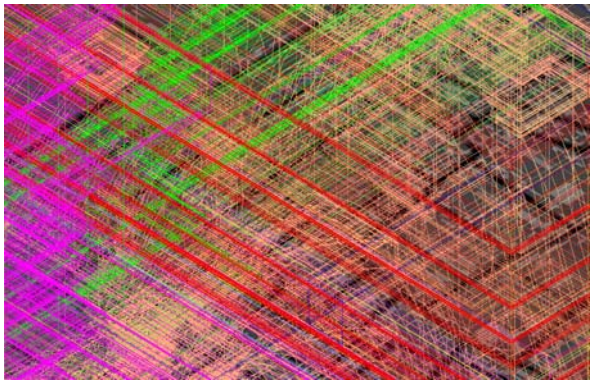


Figure 4.1.1
Wireframe view of a part of the 3D model of Angkor Wat complex in AutoCAD.

On the other hand, AutoCAD is already widely deployed to assist the architectural work for reconstruction of Angkor temples. Base on our experience in modeling as well as programming within AutoCAD, we expect that ATG can be well integrated to work environment especially to users who are already familiar with AutoCAD.

4.1.1 VBA IDE and AutoCAD

VBA is an implementation-oriented version of the event driven programming language Visual Basic (VB). Microsoft does not provide VBA only to the family tools of Microsoft Office, but also licensed it to several other applications including AutoCAD by imbedding VBA IDE in that host application (see figure 4.1.2). Although VBA is somehow limited in terms of controlling AutoCAD features compared to Visual LISP and *Object ARX (AutoCAD Runtime Extension)*, base on some criteria it is still the best choice for our project and ATG.

Because VBA compiles the codes real-time as we type, it immediately detects syntax errors and requires very less time to compile the whole program. This is of great advantage for testing the codes during design-time. The most outstanding feature of VBA is the rich resources for developing tools with complex graphic interface, and that is relatively simple for both beginners and advanced developers of customizing AutoCAD.

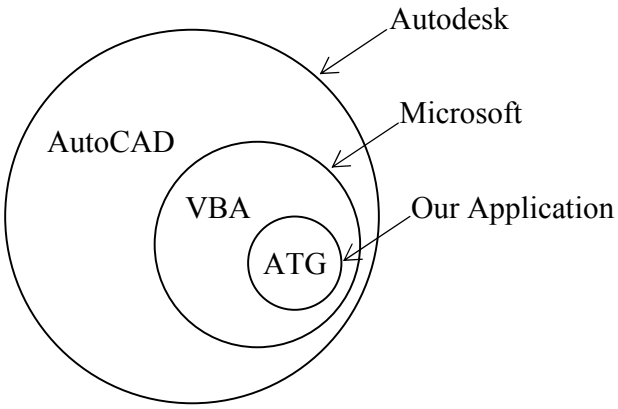


Figure 4.1.2
Hierarchy of the interactive systems of ATG, VBA and AutoCAD.

The main purpose of providing an integrated development environment in a software tool like AutoCAD is to automate specific operations within that tool. The performance of automation is generally a combination of certain available objects and functions that can be derived from AutoCAD or can be a new user generated mathematical and algorithmic expressions.

There are several ways to load VBA environment from AutoCAD. We suggest either using AutoCAD toolbar: *Tools* → *Macro* → *Visual Basic Editor*, or we can load it using AutoCAD *command prompt* by typing: *vbaide* ↵. This will generate and load a new blank project. In order to edit or explore an existing project such as that of ATG, we need to load that project in advanced. We again go to toolbar: *Tools* → *Macro* → *Load Projects…,* or using command prompt by typing: *vbaload* ↵, then from the *Open VBA Project* form window browse for the project file (eg. **ATG.dvb**), and click the *Open* button. We then can load VBA IDE with the required project (see figure 4.1.3). For more details on loading, saving and managing projects we suggest [Sut04] or [Cot01].

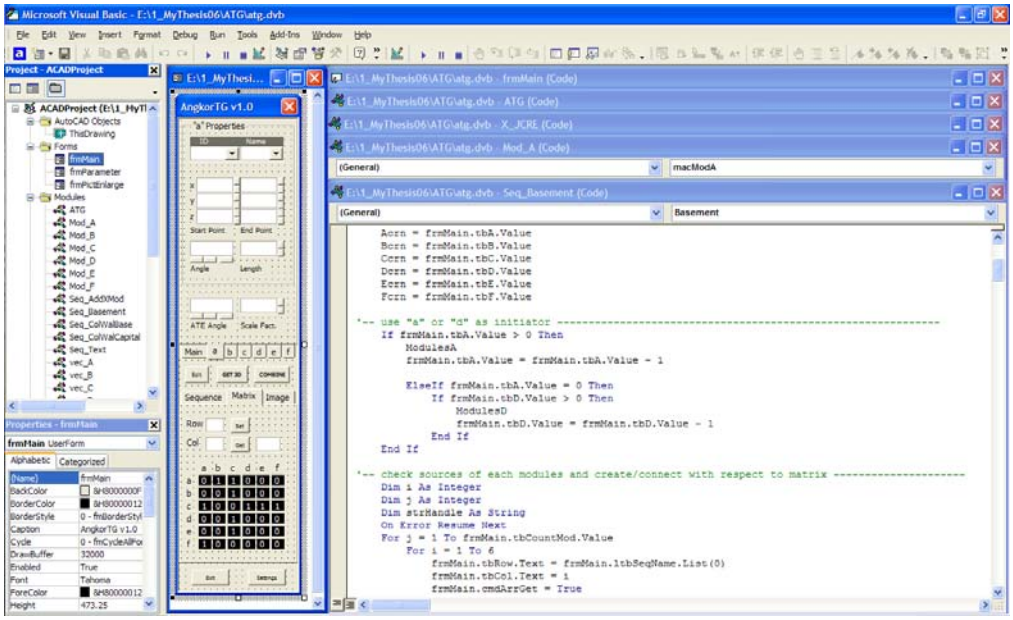


Figure 4.1.3
VBA IDE presented with ATG project and some of its components.

4.1.2 Loading and Executing the Program

VBA applications including ATG cannot be executed without loading its VBA project in advance. There are three ways to load the project. The first two options use AutoCAD

menu tool. In AutoCAD menu select *Tools* → *Macro* → *Load Project*…, then from the Open VBA Project popup window browse for the project file ATG.dvb, and open it. Or choose *Tools* → *Macro* → *VBA Manager*..., then from the *VBA Manager* popup window click *Load* button to open the Open VBA Project window, and proceed as previously described. After the project is loaded, click *Close* button to close the VBA Manager window. Option three uses the command prompt. Type *vbaload* ↵ in the command prompt, then the Open VBA Project window appears for browsing the project just the same as in previous options.

Now that our project is loaded, VBA programming environment is active and ready for controlling AutoCAD. We now can run ATG. AutoCAD provides two possibilities to run VBA projects. For user who is familiar with using menu chose *Tools* → *Macro* → *Macros*…. The window in figure 4.1.4 appears. From the macro list select ..\ATG\atg.dvb!ATG.ATG, then click the *Run* button to run ATG macro. Now you should see ATG main form loaded shown as in figure 4.2.1. The second way to run ATG is designed for user who prefers command prompt controls. In the command prompt type *-vbarun* ↵, then we are instructed to enter the macro name to run, thus typing *atg* ↵. Of course it is possible to run *Angkor Temple Detector* (*ATD*) directly without opening ATG main form in advance. This can be done following the same way we described for running or loading ATG main form but we must select ..\ATG\ATG.dvb!ATD.ATD from the list in the *macros* window or type *atd* ↵ in the command prompt.

We are now ready to have a closer look on how to use ATG and ATD. As we have seen from earlier chapters that we propose three possibilities for reconstruction of ATEs. The first method which is based on ATE modules is built on ATG main form, thus that form must loaded in advance. Other two is based on input images which requires ATD interface to be loaded. The following section demonstrates the reconstruction of ATE using the first method.

In the next section we will look closer at the physical structure of ATG in form of a pipeline. We will provide an overview on macros, and what roll it plays in the system of ATG.

4.1.3 Macros and Pipeline of ATG

Macros are groups of Visual Basic scripts that can perform specific tasks of application at the parent level application which is in this case AutoCAD. A macro can be very simple like the macro we have defined the name **ATG**. Its job is simply loading the main form **frmMain** that provides further accesses to all other macros of ATG application by clicking or modifying its controls. Visual Basic script in figure 4.1.4 (a) shows the whole contents of **ATG** macro. However some of macros are much more complex.

ATG consists of 46 project modules:

- **4 form modules:**
  - **frmMain:** Main form of ATG
  - **frmDetec:** interface for detection of element profile from image.
  - **frmSettings:** form to display options for setting features.
  - **frmPictEnlarge:** for enlargement of sample images of element.
- **32 standard modules**
  - **ATG:** for loading main form.
  - **6 modules (Mod_A** to **Mod_F):** for creating the six different ATE-module profiles (*Polyline* object) and their properties.
  - **6 modules (Vec_A** to **Vec_F):** for creating vectors (*Line* objects) of the ATE-modules and their properties.
  - **Seq_Basement:** for reconstruction of the basement vector sequence.
  - **Seq_ColWalBase:** for reconstruction of column-base, wall-base and LPE vector sequence.
  - **Seq_ColWalCapital:** for reconstruction of column-capital, wall-capital and UPE vector sequence.
  - **Seq_AddXMod:** adds the AT-modules to the corresponding vectors in *Modelspace.*

- **Seq_Text:** add module identity (*text* object) to corresponding vectors.
- **X_Clear:** clearing current list boxes before make new reconstruction.
- **X_PathCol:** create a quadratic extrusion path for the column.
- **X_PathBaseWal:** for creating a sample extrude path for the basement, wall-base and wall-capital.
- **X_PathCrown:** for creating a circle extrude path for elements such as the crown, cylindrical column, colonette or window grids, etc.
- **X_JCRE:** (Join, Close, Rotate and Extrude) for joining all generated modules, closing module sequence, and finally rotating the closed 2D profile (see section 3.4.1).
- **X_Delete2Ds:** deleting temporary generated vectors and naming texts of AT-modules to clean up *Modelspace* before 3D reconstruction.
- **X_Viewports:** create multiple *Viewports* that is recommended for 3D reconstruction using ATG.
- **Det_Delete2Ds:** deleting 2D lines created after detection of element profile.
- **Det_JCR:** joins, close and rotates the generated line sequence of element profile.
- **Det_MinMax:** controls the size of frmDetec either mini- or standard format.
- **Det_ScaleSeq:** scaling the vectorized 2D profile of element to the required dimension.
- **Det_Vectorize_LineDect:** vectorizing the captured parameter (image segmentation) of the element from image in AutoCAD.

- **Det_Vectorize_ModCapture:** vectorizing the captured parameter (by capturing modules from image), and add ATG module to the corresponding vectors.

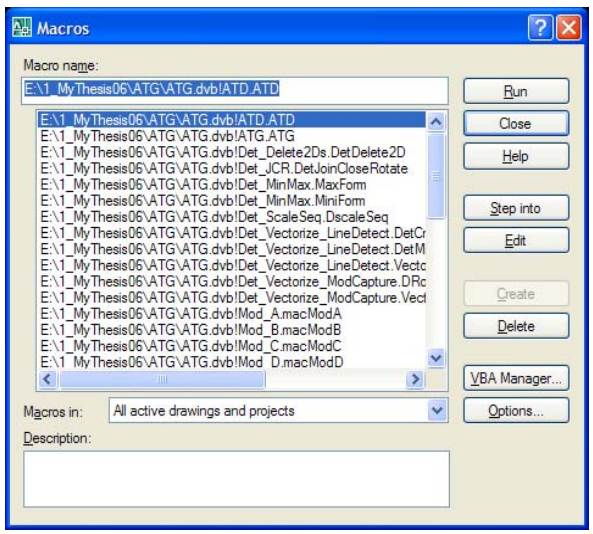
- **10 class modules**
  - **6 modules (clsMod(x)vector):** to holding properties of the six different AT-module vectors at *modified* event procedure.
  - **clsModText:** holds properties of each ATG generated texts at *modified* event procedure.
  - **3 modules (clsArray, clsArrCol, clsArrRow):** to control the adjacency matrix.

Each macro can be retrieved and run individually from AutoCAD *Macro* windows: *Tools* → *Macro* → *Macros…,* or from the command prompt type: *vbarun* ↵ then enter the macro to be run by name. The macro window appears with the available macros in the project we have loaded as shown in figure 4.1.4 (b). From this window we can select a macro and click on *Run* button. However, in order to access all feature of ATG it is necessary to load, and work on the two main graphics interfaces of ATG.

Figure 4.1.4
(a) A macro namely *ATG* performs the loading task of the main form of ATG. (b) ATG macros viewed in the Run-Macros dialog box of AutoCAD.

(a)
```
Public Sub ATG()
   frmMain.Show
End Sub
```

(b)



We have seen in chapter 3 that ATG provides three alternatives for reconstruction of Angkor temple elements. The first is based on construction rule of Khmer temple architecture, and the last two are based on available photography of the elements. ATG is design according to this subdivision. The architecture of the three systems is partially in common. Let us firstly have a look at the pipeline of the system that is based on the construction rule. Compared to the rest, it has the most complex structure as shown figure 4.1.5.

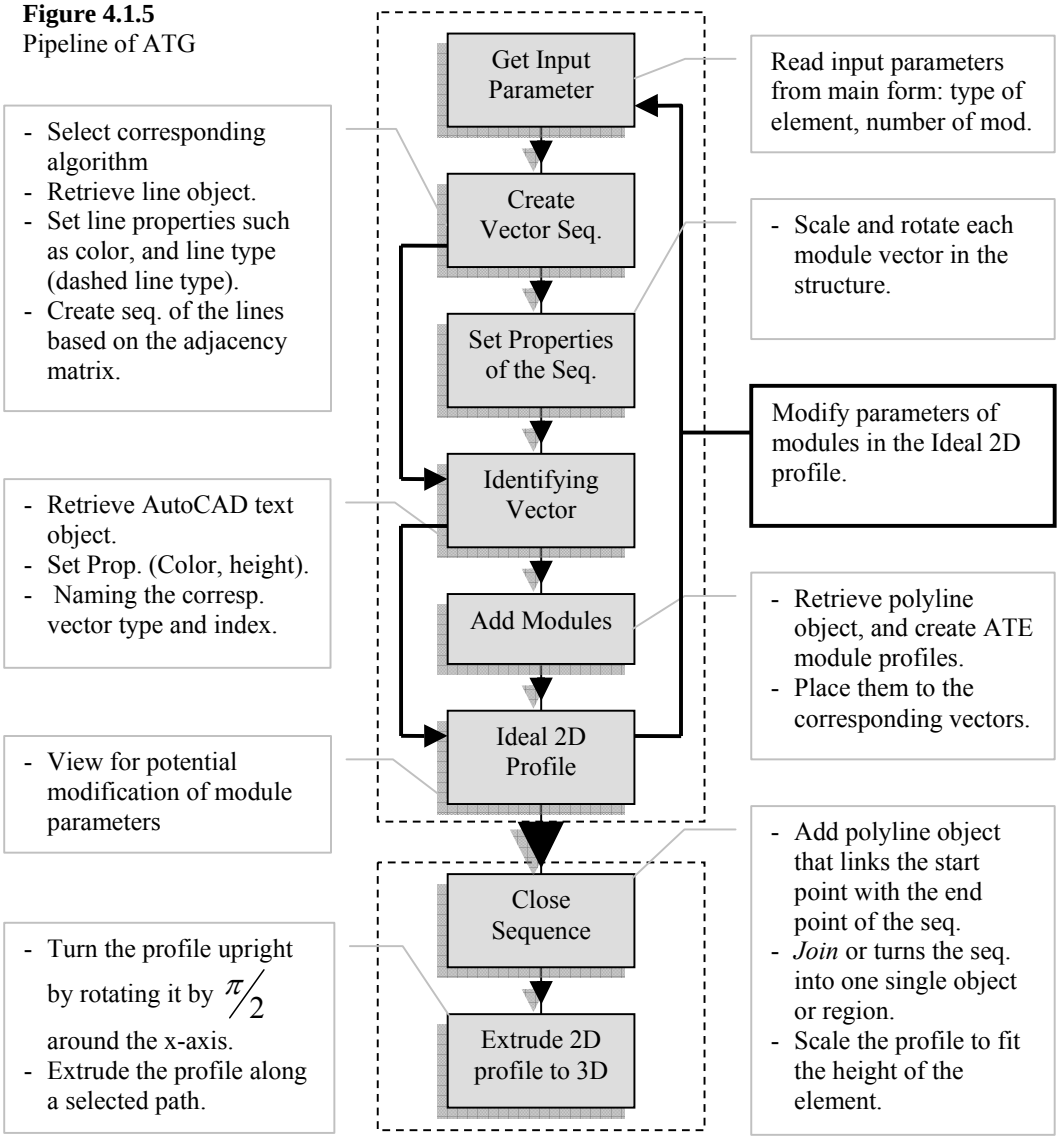
At first, the system reads the parameters necessary for reconstructing a given type of *ATG-ELEMENTS* on the main form of ATG including the number of modules of each type (see also figure 4.1.6). Then it select the combination rule based on the element type (special case: basement and UPE) before deriving the required objects from AutoCAD, in this case a *line objects*, and sets the corresponding properties including line type as dashed line and the corresponding color type which distinguishes types of module vectors as predefined in ATG library. Each line is to be linked to each other with respect to the construction rule forming up a horizontal sequence of homogeneous module vectors.

In step three, the system adjusts the proportion and angle value of each component of the element structure. The fourth step makes the reconstructed module vector structure more readable. Here ATG derives *text object*, placing it at the bottom right of the start point of a module vector, and labeled it by the identity of the corresponding module type as well as the indices (of its type) in the structure. Now that we know all relevant properties of each component in the element structure, the modules (module profiles) can be added to the corresponding vectors in the structure.

Up to the fifth step, all processes are done automatically with just a single click on the *COMBINE* button of ATG's main graphics user interface. We obtain a complete ideal profile of Angkor temple element. In some cases we require to adjust the properties of the resulting element profile. At this phase we are still able to modify parameters of each module. In such case, the system simply begins the reconstruction procedure (from step one to four) as described above anew to reproduce the structure after the newly adjusted parameters.

In step seven, ATG retrieves another polyline object from AutoCAD to *close* the polyline (module) sequence of the element structure. This operation converts all modules into one single *closed* polyline object or *region*. After this phase, ATG scales the ideal dimension of element profile to fit the height of the element to be reconstructed.

Figure 4.1.5
Pipeline of ATG



Finally in the last step, the system rotates the closed 2D region upright so that the plane of the region is positioned perpendicular to the plain of the extrude path. Then it

asks the user to move the region and place it at the proper position in relation to the extrude path, and extrude the profile along that path as it is selected by a mouse click.

4.1.4 User Interface and Features

The compactness ATG graphics form (160 x 650 pixels) preserves large work area or *viewport* of AutoCAD for user to operate, and provides access to features of ATG and AutoCAD easily. This is necessary because ATG depending on AutoCAD modeling tools and operations such as drawing objects, text objects, viewer etc. Generally if a *graphics user interface* (GUI) form of an application is in *modal* mode, the other forms related to the same parent application or the parent application itself either must be closed or must be set to inactive which means no access allowed until the active one is closed. Unlike *pallets* and other controls which are parts of AutoCAD controls, we are constrained to design ATG on a separated *form module* which makes the switching from between ATG and AutoCAD GUI a little bit inconvenient though not impossible.

As ATG is loaded, its form is set to *modeless.* This feature enables accessibility to both forms; ATG and AutoCAD where AutoCAD has priority. This means that while we are working on AutoCAD, ATG's form remains on top of AutoCAD, and can (only) be accessed by mouse click events on its controls, and is automatically set back to modeless after releasing the mouse button. A minor drawback is when you need to enter or modify parameter by keyboard, for instance in a text box, you need to click the *Edit* button to turn ATG form to modal, and turn it off when finish.

The architecture of ATG is built in such a way that the user can easily access most frequently used components including the *taps* of each type of module, the main *tap* and others components, marked with the red dashed lines in figure 4.1.6., by designing them close to each other on the main form.

The second important graphics interface is the form of Angkor Temple Detector which can be loaded by clicking the *ATDetector* button on the Main tap of ATG main graphics interface or using command prompt as will be shown in the next section.

Figure 4.1.7 presents the graphics user interface of ATD; one for capturing module properties, and the other being in the process of detecting the profile from an image of a column capital of Angkor Wat temple. Unlike ATG, ATD interface must be designed larger (1000 x 650 pixels) since it deals with digital images as input data for reconstruction of ATE profiles. It is therefore also a little bit more inconvenience to interact with AutoCAD interface. Also on this form we implement the switch option between modeless and modal modus, and simultaneously kept on top of AutoCAD application together with ATG form.

The rest two forms are classified as second-rated. One of them is used to hold settings parameters, and can be loaded by clicking the *SETTINGS* button on the lowermost part of ATG main form. The other form holds the enlarged images of sample drawings of ATE to be reconstructed, and appears when clicking on the toggle button with the magnifying glass icon on the *Image* tab.

Before we demonstrate reconstructing Angkor temple elements using ATG and ATD, we need to know how to run them as being a child application of AutoCAD. In the last part of this section we explain what conditions are required in order to execute these software tools.

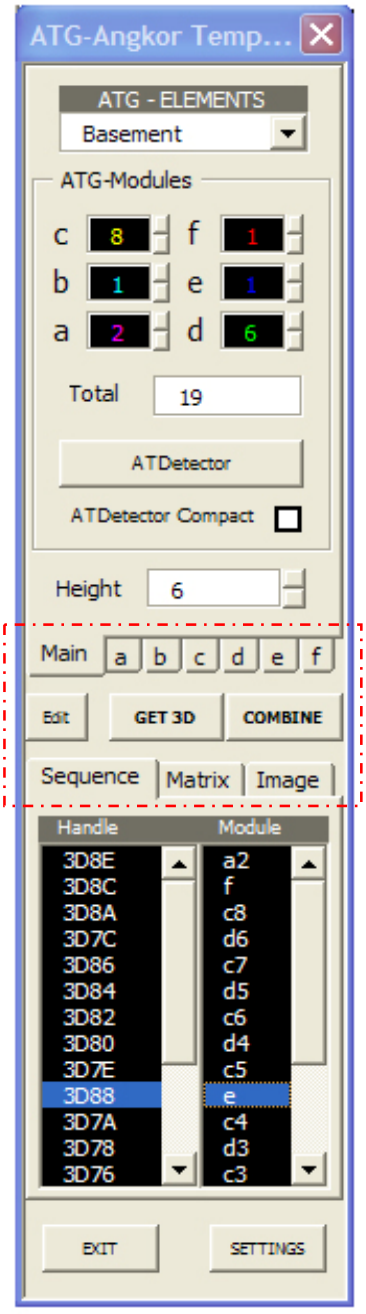


Figure 4.1.6
ATG main graphics interface.

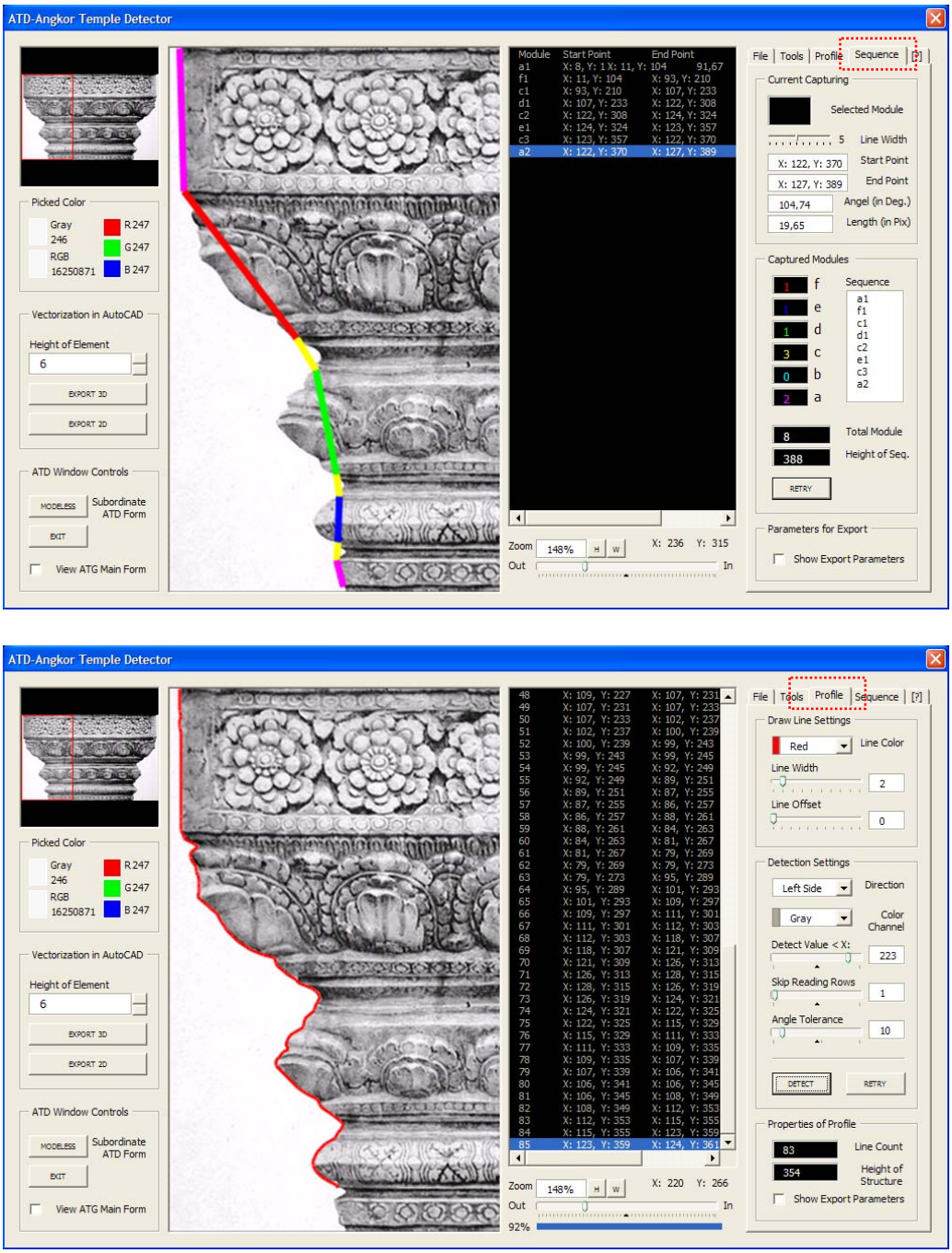


Figure 4.1.7
Top: *Sequence* tab, a tool for module sequence reconstruction by capturing module properties from a digital image. Bottom: *Profile* tab, another tool to detect element profile from a digital image.

4.2 Module-Based Reconstruction Using ATG

In this section we discuss about the functionality of ATG. We begin with input data required for reconstruction an ATE, and how they are managed in the reconstruction process. Section 4.2.2 presents the management of ATG objects and data structure generation. Finally we introduce some special features of ATG for controlling AutoCAD features to gain advantage for the reconstruction with a reconstruction demonstration.

4.2.1 Input Parameters

Common approaches for modeling objects with comparative similar geometric complexity as ATE require an image of the objects used as input data to manually vectorize the profile. Another method is to measuring every geometric detail which is in practice very limited because of the difficulties in both tasks; measuring and modeling. These processes must be done manually, and are the most time consuming tasks in 3D modeling.

In contrast, with the advantages of the predefined algorithms and objects, ATG requires only two fundamental parametric settings:

- Defining element type to be reconstructed in order to identify the associated algorithm
- Numbers of modules of each type needed as the composition of the structure.

Reconstruction of a basement for instance, we simply need to set the *ATG-ELEMENTS* in the combo box on ATG main form shown in figure 4.1.6 to *Basement* and required number of each *ATG-Modules* right below it. Then a complete ideal profile can be reconstructed with just a single click on the *COMBINE* button. Figure 4.2.1 shows an example for reconstructing a basement profile, and the modeling environment of ATG with AutoCAD.

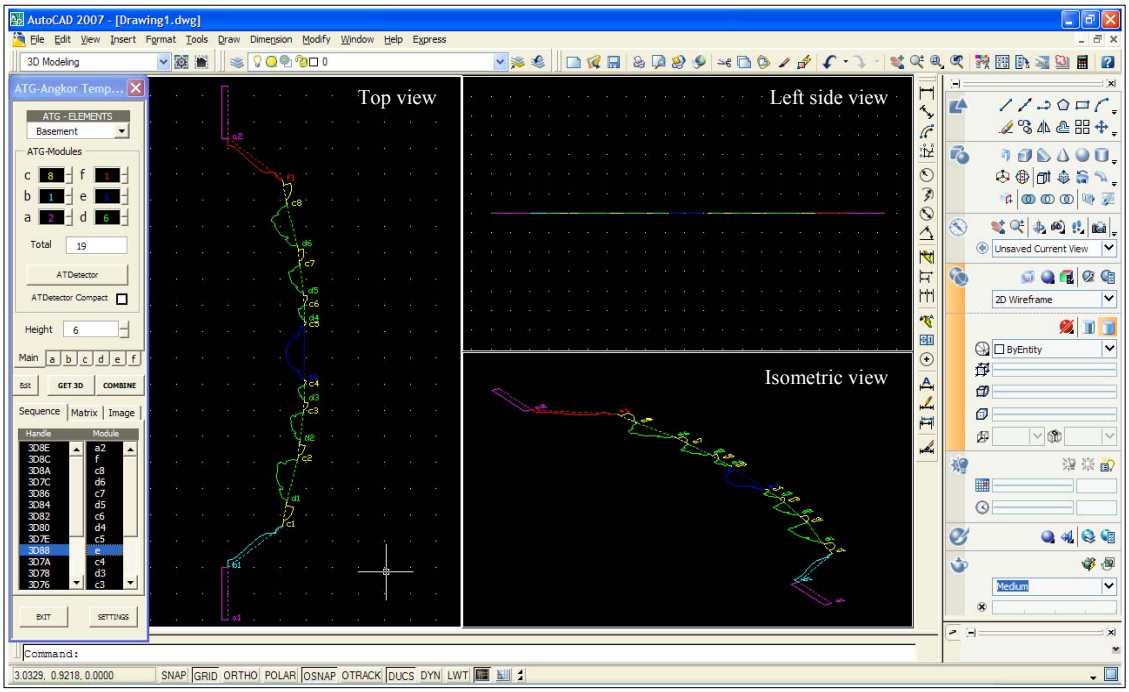


Figure 4.2.1
ATG and AutoCAD modeling Environment, and a sample for reconstructing a basement profile.

More detailed description on modeling using ATG is discussed in section 4.2.3. Let us firstly have a closer look at the data structure ATG produces based on these input parameters, and its management of AutoCAD objects.

4.2.2 Object Representation and Data Structure

All objects of any types including 3D object, line, polyline, text etc. that have been generated in the *Modelspace* of AutoCAD is identified by individual object identity called object *handle.* The name consists of letters or numbers or a combination of both for instance *AEC, 2268, 3C0*, etc. and is automatically defined by AutoCAD system.

In order keep full control of all objects that corresponds to ATG it is necessary to generate data structure in form of ordered lists of object arranged by object type and module type. Above all we must create an adjacency-list of module vectors (line objects) that represent the structure of the temple element to holding the handles of all objects.

Figure 4.2.2 (c) shows three lists of objects in *c*-module group including module profiles (polyline), module vector (line) and the naming object (text). The list of module vectors is the most important amongst them because other related objects are always referred to the line objects. Moreover, as we have discussed in section 3.1.3, the angles and lengths settings of *c* and *d*-module are defined in regarding their indices in a particular group, thus we need this list to store them in an associated order as they are initialized as well as to provide access to the objects for further modifications.

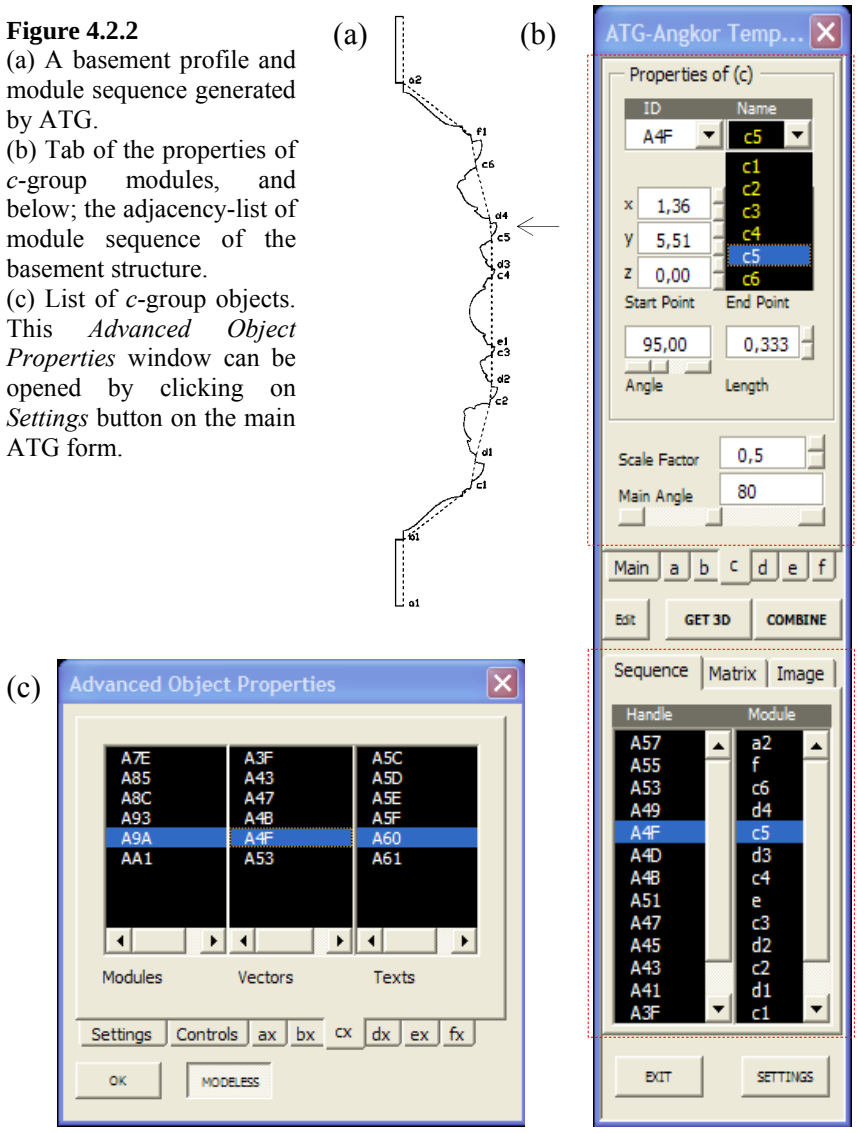


Figure 4.2.2
(a) A basement profile and module sequence generated by ATG.
(b) Tab of the properties of *c*-group modules, and below; the adjacency-list of module sequence of the basement structure.
(c) List of *c*-group objects. This *Advanced Object Properties* window can be opened by clicking on *Settings* button on the main ATG form.

With the object handle we can get and set relevant information of the object property. For `AcadLine`, a class for AutoCAD lines objects used for VBA for instance, there are following available properties:

- `Handle:` Get identity of the object
- `Angle:` Get angle at X-axis proceeding counterclockwise
- `Delta:` Get delta of X-, Y- and Z-axis
- `EndPoint:` Get and set coordinates of its end point
- `Length:` Get length
- `Normal:` Get and set normal
- `StartPoint:` Get and set coordinates of its start point
- `Thickness:` Get and set it distance extruded above or below its elevation.

ATG provide access to relevant properties of each objects for the case of individual modification as for the module *c5* shown in image (a) and (b) of figure 4.2.2. For more detailed information for programming AutoCAD objects and their properties see [Sut04].

Another very important data structure is stored as the adjacency-list of objects consisting in the temple element structure. Here again we store their handles with a link to the associated object names as shown on the *Sequence* tab of ATG main user interface of image (b). ATG is designed for an experimental purpose to observe and analyzing the reconstruction process. We can observe from software pipeline that it firstly generate the chain of module vectors. The next step however, scales and rotates the initialized component with specific factors that breaks down the sequence into individual components. At this step we use the adjacency list to reconnect them in their current order.

In relation to ATD capturing tools, this list is used to associate the parameters of module properties that have been captured from an image to the objects to be vectorized. We will see the practical use of this term in the next section. After all, this list is also useful for printing the results of reconstruction or capturing process.

4.2.3 3D Reconstruction and Special Features

ATG provides a new dimension of 3D computer reconstruction of Angkor Temple. The very limited requirement of input data and the specific algorithm designed following the Khmer temple construction rules make possible to reconstruct a complete 3D model of the temple element with just a few mouse clicks.

More advantageous features of ATG that we must have mentioned is that it contains of specific repertoire of commands for controlling selective 2D and 3D operations of AutoCAD system. Each collection of commands automatically executes a routine of operations for a particular purpose, and is either imbed in a code *sub procedure* or can be accessed by mouse click event.

For instance the *GET 3D* button executes the following:

- Automatically add the three line segments we have discuss in section 3.4.2, *join* all polyline objects in data structure to convert the whole structure into one region
- Scale the element profile to fit the required height of the 3D model
- Rotate the profile upright from the ground floor plain
- Extrude the profile a long a given path.

All these operations are automatically executed except the selection of the support origin *O* on the profile and selection of the corresponding path which should be done manually. A demonstrate of module based 3D reconstruction using ATG is given a following.

For the reconstruction demonstration we choose the following input data:

- Temple element type: basement
- Components: $S_{basement} := (2a, b, 6d, e, f)$
- Height of basement: 3 unit
- Define the ground floor projection of the basement (extrude path)

That is all we need as input parameters. Figure 4.2.3 shows a demonstration of the reconstruction procedure which is divided into seven steps. The time required to precede the whole process is about few seconds only.

Figure 4.2.3
Reconstruction of a basement with ATG.

1. Select element type: the basement.

2. Set number of each component.

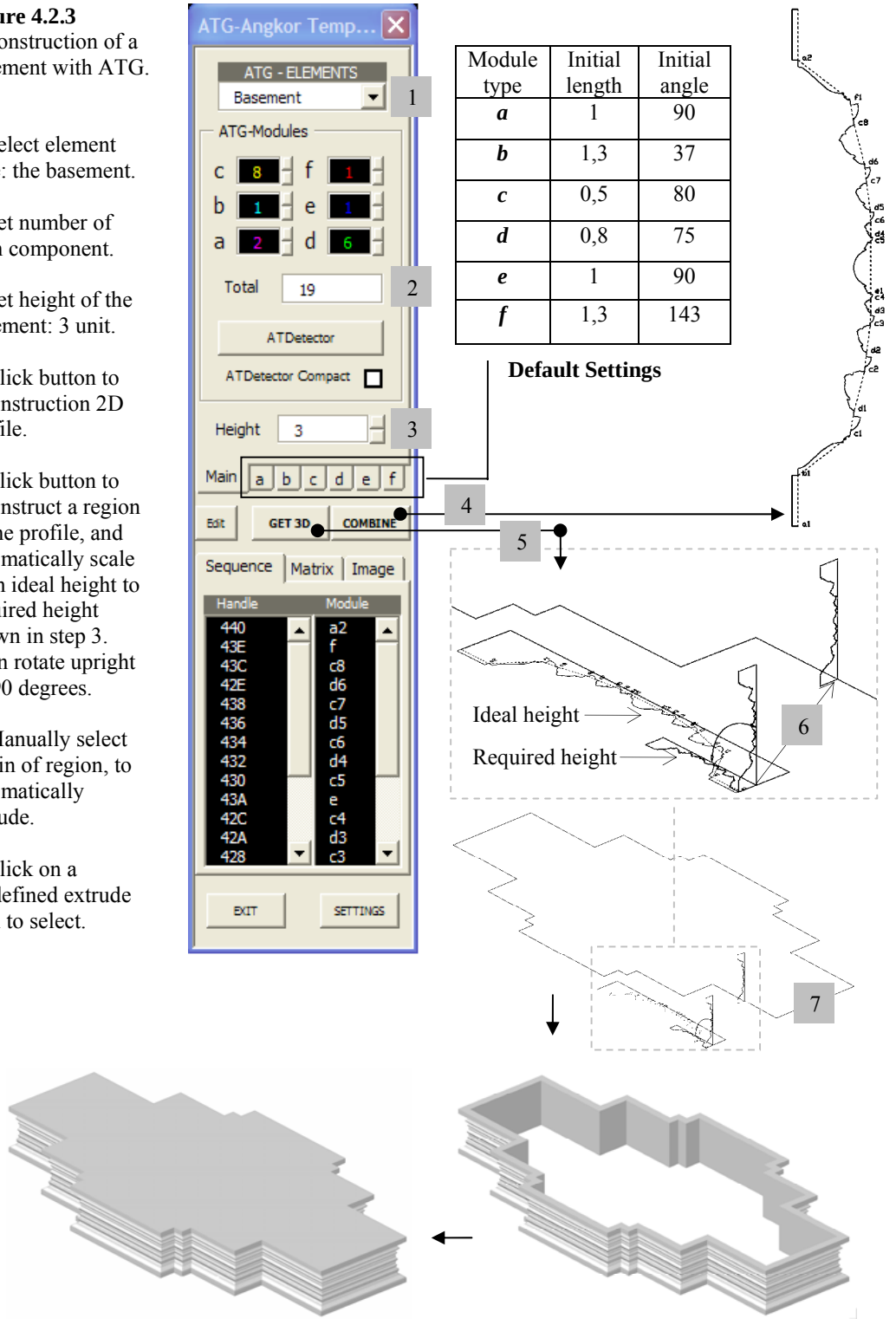
3. Set height of the basement: 3 unit.

4. Click button to reconstruction 2D profile.

5. Click button to reconstruct a region of the profile, and automatically scale from ideal height to required height shown in step 3. Then rotate upright by 90 degrees.

6. Manually select origin of region, to automatically extrude.

7. Click on a predefined extrude path to select.



Module type	Initial length	Initial angle
a	1	90
b	1,3	37
c	0,5	80
d	0,8	75
e	1	90
f	1,3	143

Default Settings

To modify the basement structure we can go to the properties tab of each module type, and change the default settings with new parameters yet before the fifth step (see also figure 4.2.2 (b).

The module-based reconstruction approach can be very well intergraded with more detailed measurements of an available ATE to reconstruct the element following its current state. This is one of the main tasks of ATD tool which we are going to present in the next step.

4.3 Angkor Temple Detector (ATD) Tool

Earlier we have introduced that ATD provides two different tools for reconstruction of ATE, which one of them are based on capturing module (module properties) from an available digital image of the element. On the other hand, the second tool reconstructs by automatically detecting the profile from the image. Section 4.3.1 we introduce an image processing class which is necessary for software development within VBA. Then in the last two parts we discuss and provide samples for reconstruction using both tools; capturing module properties tool and detection of ATE profile tool respectively.

4.3.1 Image Processing Class for VBA

Developing software tools for controlling AutoCAD using VBA is almost unlimited as described by [Cut01] and [Sut04] however, it is very limited when dealing with images and image processing functions. VBA does not comprise a *picture box* control from which properties of picture elements (pixel) of an image can be examined or modified. Instead an *image box* is available for loading an image only.

ATD is designed with an image processing ActiveX control called *GDPicture Pro* which can be used in various developing environments including Visual Basic, Visual FoxPro, MS Access Delphi, VB.NET, C#, Visual C++, RealBasic, etc. GDPicture Pro is a product of a young company called *gdpicture.com,* founded in 2003 and focusing on

developing imaging toolkits. For more information we suggest to visit their website: www.gdpicture.com.

Our main purpose to use GDPicture is to access each pixel of an image of a temple element from AutoCAD in order to examine the picture properties, and thus the properties of the temple element. Besides, GDPicture provides many useful features for handling the image such as zooming, rotating, image quality display, etc. Instructions on adding the control, system registry, programming reference, and other technical topics can be found in the documents coming with the toolkit.

We have seen from chapter 3 that the modules capturing tool of ATD does not require to examine any pixel of the input image however, we use it to capture coordinates of certain selected pixels. In the following section we present how this works.

4.3.2 Capturing Module Properties

We have already mentioned how to load the graphics user interface of ATD in section 4.1.4. This interface is necessary to be loaded in order to work with ATD tools. From the *File* tab we can open an input image, which must be a RGB color channel image because we have defined color line objects to trace the captured module vector.

As we have identified the first module (topmost module) on the image we must firstly insert the module type into the system using the keyboard. Then with the mouse cursor we select on the input image each node of the module sequence in the top-down directional order. **Remark:** as a rule, we only need to select the start points of each selected module. The start point of the preceding captured module is automatically defined as the endpoint of the module to be captured in order to guaranty a continuous chain of module vectors. But for the initial module, both ends must be picked.

For this reconstruction method we again need to generate data structures of the captured element structure. Relevant parameters are the following:

- Module sequence of the whole element structure
- Angles and lengths of the modules in the element structure

Theoretically it is sufficient to generate data structure of coordinates of the nodes which are represented by the coordinates of the corresponding captured pixels to reconstruct the element profile structure. But ATD calculates the lengths and angles of each module real-time during capturing process from the captured coordinates. Doing this, we obtain a more informative data of the modules for printing for instance. The modules are vectorized in AutoCAD using the data structure of the computed lengths and angle values. To vectorize the results, we simply need to click on the *Export 2D* button on the left hand side of ATD graphics interface, see figure 4.3.1.

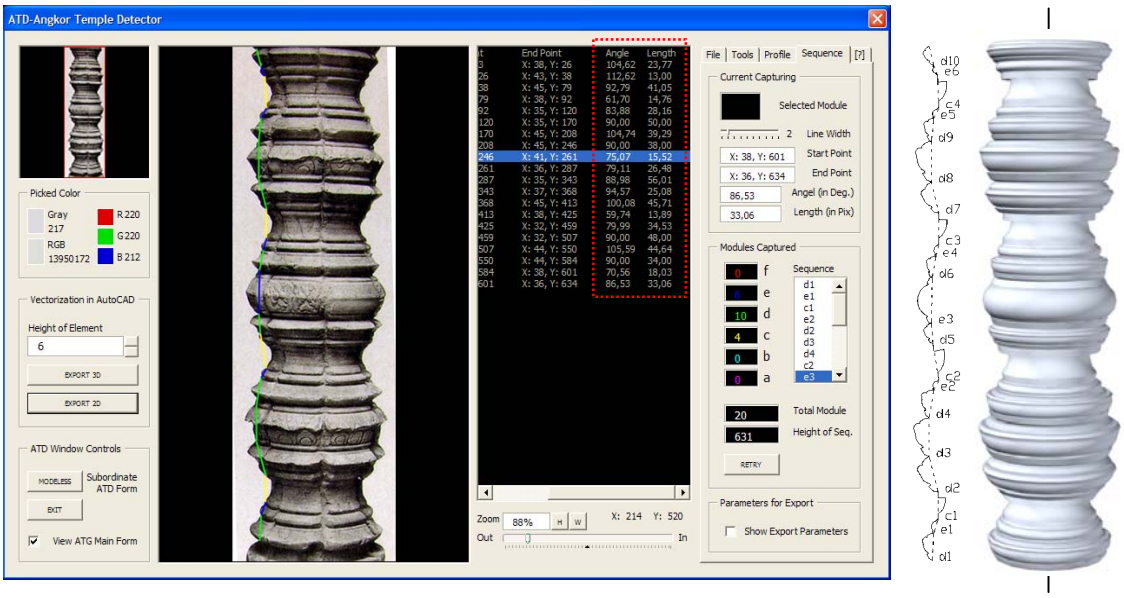


Figure 4.3.1
Left: Capturing module structure of a part of a window gird of Angkor Wat temple. Middle: Reconstruction of the captured element profile. Right: 3D model we finally obtain.

This figure also emphasizes the main advantage of using ATD capturing tool. We observe that we can freely connect distinct module type with each other in case of reconstructing special element structure which original structuring concept is apart from our construction rule. On this window grid element we find a sub-sequence of three *d*-modules, and *d*-module is directly linked with an *e*-module, for instance which is not defined in the reconstruction rule of ATG.

Although the module capturing tool of ATD only provides a half-automated reconstruction method, it is crucial to have flexible options of combining distinct module types. After all it is still far faster if compared with conventional modeling methods to reconstruct complex element structure such as the window grid. For the reason that it employs ATG predefined module profiles, we finally obtain a highly detailed 3D model.

The following section we present a full-automated approach for detection of temple element profile from a digital input image.

4.3.3 Detection of ATE Profile

The temple element profile detection tool of ATD, unlike the other two methods, is not a module based reconstruction tool. Together with the module capturing tool, it is built on ATD user interface but in a different tab called *Profile* tab. Here we find various control options including the both levels of detail controls presented in section 3.3.3, see figure 4.3.2. The parameter *T* of the angle deviation tolerance is defined here using the slide bar *Angle Tolerance* in the *Detection Settings* frame. Above it we see the slide bar *Skip Reading Rows* for setting the number of pixel rows to skip for each detection interval.

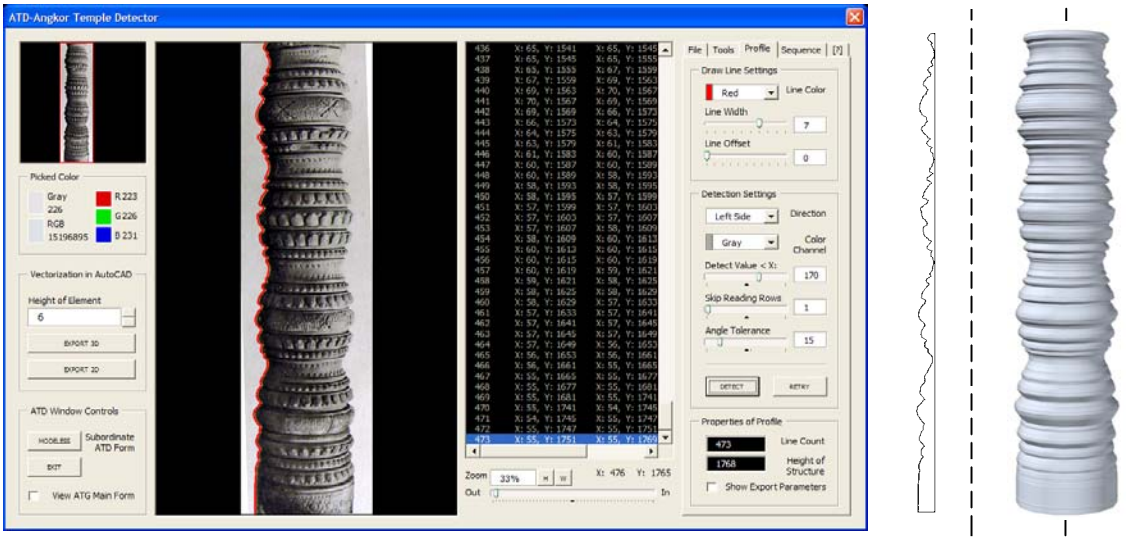


Figure 4.3.2
Left: Detection of a very detailed window grid of Angkor Wat temple. Middle: Reconstruction the detected profile. Right: 3D model of the window grid.

It is also important to mention the function of *Detect Value < X:* slide bar. As default its value is set to 255, the maximum gray value of a pixel. We have seen in section 3.3.1 that for the case that satisfies the inequality $g^P < g^B$, the pixel E^P is considered as a pixel of the element profile. Thus it is important to set the X value properly. If we define it too large ATD will register any uneven pixels on the background, but if it is set too low the system will partly ignore the threshold. Figure 4.3.3 presents an example on setting three different tolerances of the gray value.

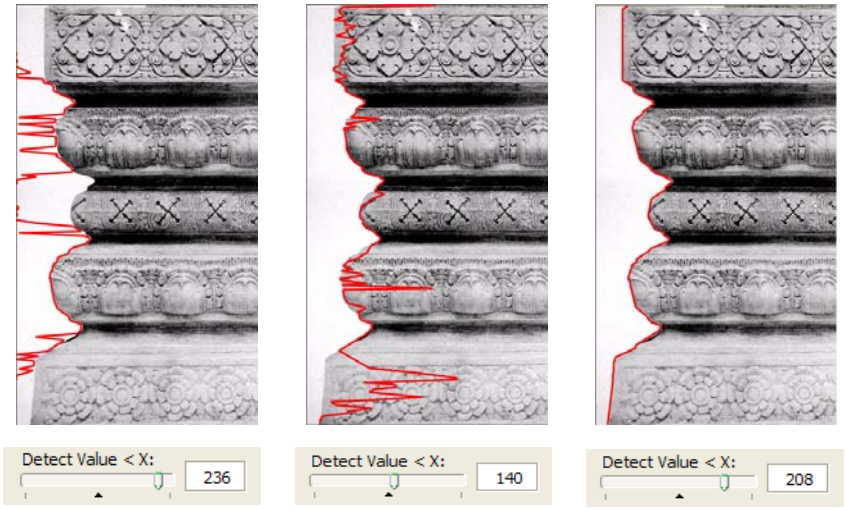


Figure 4.3.3
Three different results each of which corresponds to a certain level of tolerated gray value setting. Left: too large, Middle: too low, and Right: proper value

In order to find out the lowest tolerated gray value of the background pixels, we have developed a function that records the gray value from a set of pixels that the user selects from the background. As an input image is loaded the initial value of $X = 255$. Thus the first task to do is locate the mouse cursor on the background area, and press the *Shift* key. With the key remain pressed down, move the cursor to the likely darkest area in the background or visible dark spot without crossing over the object (temple element). We observe that the value of X reduces as we move the cursor to a darker area.

A well shaded photography of the temple element require no additional modification to *X*. Otherwise we might miss a few dark spots that can cause minor detection error but it can be solved by reducing *X* by a few levels or use ATD image editor tool in the *Tool* tab. Thus to use this tool effectively it is necessary to use good quality input image with following relevant properties:

- High contrast between object and background
- Prevent distortion: image should be made from large distance.

Finally, to detect the element profile we must click the *Detect* button.

Unlike the module capturing tool, the data structure generated by this system consists of coordinates of the detected pixels which will be directly exported to AutoCAD without additional calculations except to convert the coordinate system (see section 3.2.2).

4.4 Conclusions and Further Improvements

ATG is a software package that consists of three different tools for a time-saving and high quality 3D reconstruction of Angkor temple elements with the main focus on reconstruction of Angkor Wat style temples. The main tool of ATG is based on a brand new technique which is the result of a combination of ancient Khmer temple architecture and state-of-the-art technologies in computer science.

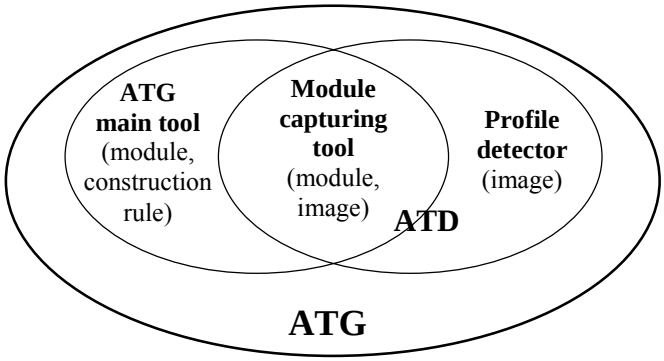


Figure 4.4.1
The three tools of ATG for reconstruction of Angkor temple elements.

The deployment of the tools depends on the available input data and the purpose of the reconstruction. Predefined modules together with the imbedded reconstruction rule make the ATG main tool to be the most user-friendly tool amongst the other two.

As a completion to it, the module capturing tool of ATD gives the user the opportunity to freely choose the way of combining the predefined modules. This tool is specified for reconstruction of non-ATEs as well as complex elements of ATE sorts. It can be well integrated with input images of temple elements of which surface is party eroded but the sequence structure could still be identified by the user.

For the case of reconstructing and existing element which is still in a good condition, and an input image of it exists, the profile detector is the appropriated tool.

Further Improvements: ATG that is attached with this book is a beta version software package, designed for experimental purposes of this research. Some unexpected errors might occur when working with this version which must be maintained in the future. More relevant subjects that we suggest to improve or further develop on ATG are the following:

- Add user-defined controls for modification of length and angle functions.
- Implement the inverse problem: automate the module shape detection from a temple element profile of an input image by finding out the most suitable match between the predefined ideal module profile and the associated module profile on the image.
- Automate the filling process introduced in figure 3.4.6 (b) to column and basement and elements with the associated volumes.
- Implement the rule number 3 of the definition of the general condition to automatically sets the number of *c*-module based on $n_c = n_a + n_d + n_e - 1$.
- Automate ATD 3D reconstruction of elements with basic geometric extrude path: column, window grids, crown, etc.

- Element profile detection: Do not begin to analyze the pixels in the proceeding row ψ_{i+1} from column index $\xi_{i+1} = 0$ but from $\xi_{i+1} = \xi_i - x$ where x defines the number of pixel columns to turn back from column ξ_i. This method will reduce the number of pixel to examine, and thus enhance the speed of detection.
- Develop a user-defined option for selecting the interval of pixel rows to be detected on the input image.
- Develop multi-level of details of the six module profiles

Chapter 5

Reconstruction Using ATG and Results

In this chapter we are going to present the results of using ATG tools for reconstruction of Angkor temples and Khmer temples. The first part of this chapter is about 3D reconstruction of Angkor Wat style temples by taking a library of Angkor Wat complex as our case study. Further results on using ATG to reconstruct the whole Angkor Wat temple complex with basic virtual reality 3D models of the library as well as Angkor Wat are presented in this section as well.

In the second part we discuss the reconstruction of element profiles of non-AWSTs using ATG tools. We divide them into two groups; the group of temples before Angkor Wat style period and after Angkor Wat style period, and study the differences of their architectural properties relative to Angkor Wat styles and ATG modules with samples of few selected elements.

5.1 3D Reconstruction of the Library and Angkor Wat

In this section we are going to present the 3D computer reconstruction of temple elements in Angkor Wat style using ATG and ATD. As a sample for our reconstruction we chose the southern library which is located on the second level of Angkor Wat temple complex. Several elements that belong to the library will be separately reconstructed which finally are composed together as whole, and is presented in section 5.1.1. At the end of this

section we will also show some more results of using our software tools to reconstruct the whole Angkor Wat temple complex with virtual reality visualizations of the library and the Angkor Wat temple.

5.1.1 Measurements and Structuring the Extrude Paths

As for our first reconstruction sample using ATG we select a single building; the so-called *Library* of Angkor Wat temple complex. In the Chronology we listed some relevant descriptions about Angkor Wat including the construction period, the dynasty, capital, etc. For historical details about Angkor Wat temple complex we highly suggest to read [Sti82] and [Jac99].

Figure 1.2.1 we have already given some pictorial description of the library as well as the reconstruction principles that we are going to discuss now in detail. Based on this figure together with figure 1.2.2 and 1.2.3 we also know that our method for 3D reconstruction of ATE requires two sources of information: one is the information for reconstruction of the element profile, and the other is information for reconstructing the extrude paths of the corresponding profile.

The major information resource we get for this reconstruction purpose is measured from the hand drawings what were made by EFEO, and published in the year 1969 [Naf69] including the ground floor plan, elevation plan and a cross section of the library. We will also use information that was gained from the photography of the temple elements in order to obtain more reliable input data. However, the extrude paths (ground floor plans of the elements) are totally based on the drawings.

Before we begin with reconstructing the element profiles using ATG, we must vectorized their extrude paths in AutoCAD in advance. This can be done by importing the ground floor plan drawing of the library into AutoCAD *viewport*, and use the polyline object to manually draw over the ground floor projection of each element.

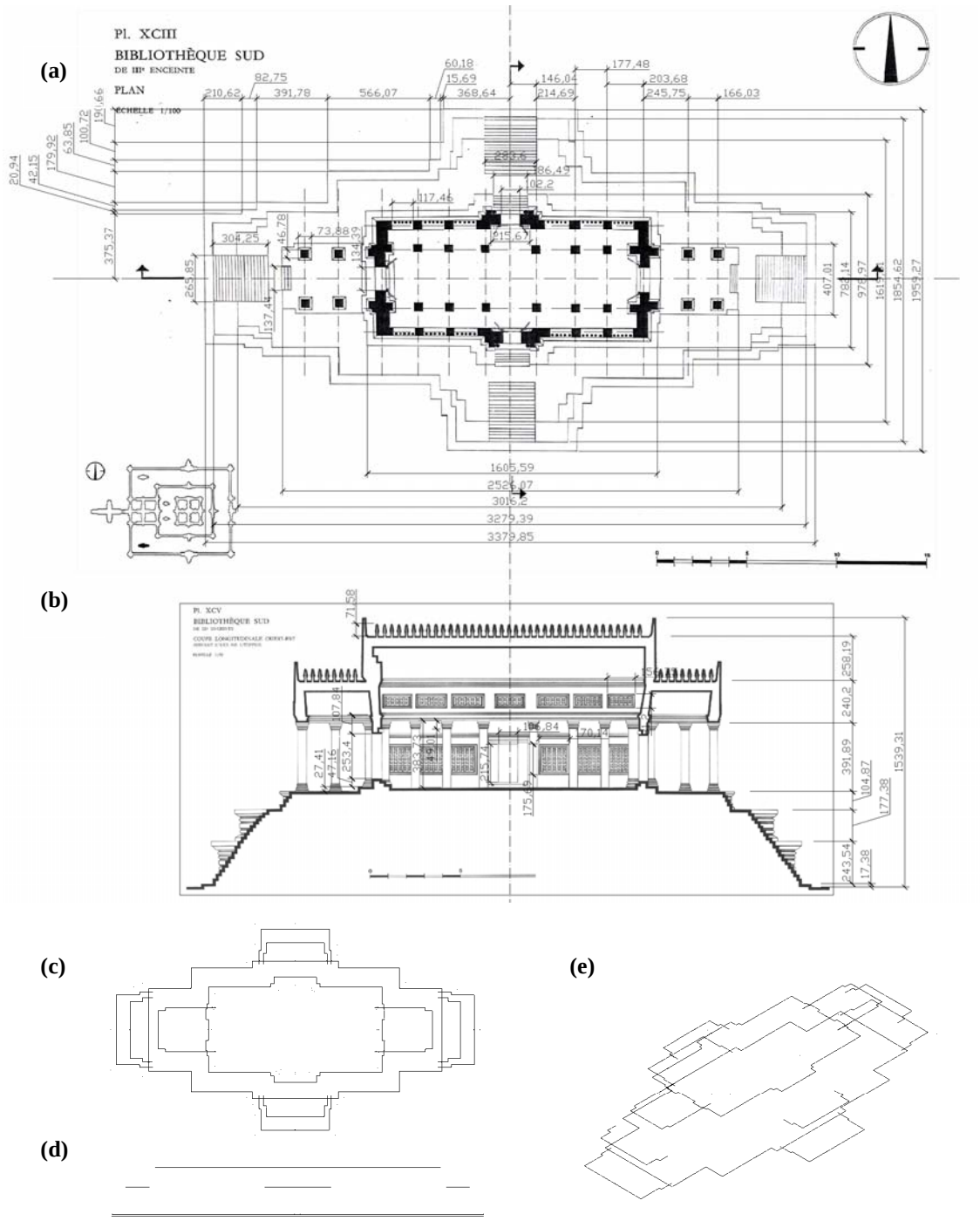


Figure 5.1.1
Measurement (in centimeters) of the southern library at the second level of Angkor Wat for reconstruction of the extrude path of a few selected elements of the library. (a) Measurement of the ground floor plan, (b) measuring the elevation, (c) vectorized version of the ground floor plan, (d) height of each path, and (e) isometric view of the paths.

We can also do measurements of all elements to get their dimension and location relative to a selected reference point which is generally the main axis of the building. Then we need to find out their locations on the *X*-*Y* plain, the height of each element as well as their elevation. The measurements we demonstrate in figure 5.1.1 where the image (a) shows the measurements on the ground floor plan, and (b), on the east-west cross section. Image (c) shows the top view of the vectorized version of the outlines of some selected elements, and (d) and (e) show their elevations and isometric view respectively. After the extrude paths of the elements are vectorized and are located in their proper positions, we can start to reconstruct the element profiles.

From these selected elements let us take a few elements to analyze and demonstrate the reconstruction methods in detail. We define the name of each element as shown in figure 5.1.2, and we start with the reconstruction of B1, and then B3, the wall base W1, wall capital W2, the column C1, and finally the window grid G1.

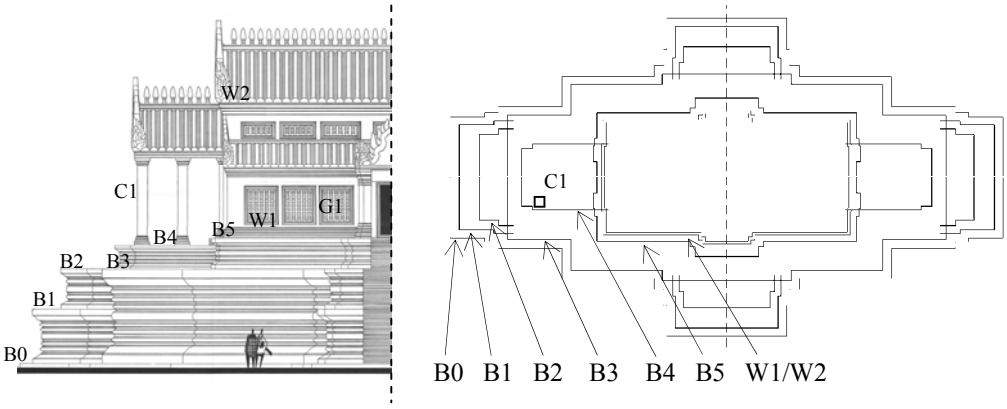


Figure 5.1.2
Left: elevation of the library and the selected elements for reconstruction. Right: the extrude paths of the elements.

Remark: In order to assemble each element model on each other properly, we must make sure that the heights of the element profiles match with the levels of the predefined extrude paths as shown in image (b) and (d) of figure 5.1.1.

5.1.2 3D Reconstruction of Elements and Assembling

For reconstruction of B1 we use the photography of the actual basement as input image in order to reconstruct the profile, and the height of the basement we get from the measurement in figure 5.1.1. We firstly try to reconstruct it using the main tool of ATG that is based on the predefined reconstruction rule and ideal proportional settings of the modules. From the image we can identify the structure of B1 as $S_{B1} := (3a, b, 4d, f)$. We give these input data to ATG, and obtain a profile with the module structure:

$$S_{B1} = ({}_1a, b, {}_1c, {}_1d, {}_2c, {}_2d, {}_3c, {}_2a, {}_4c, {}_3d, {}_5c, {}_4d, {}_6c, f, {}_3a).$$

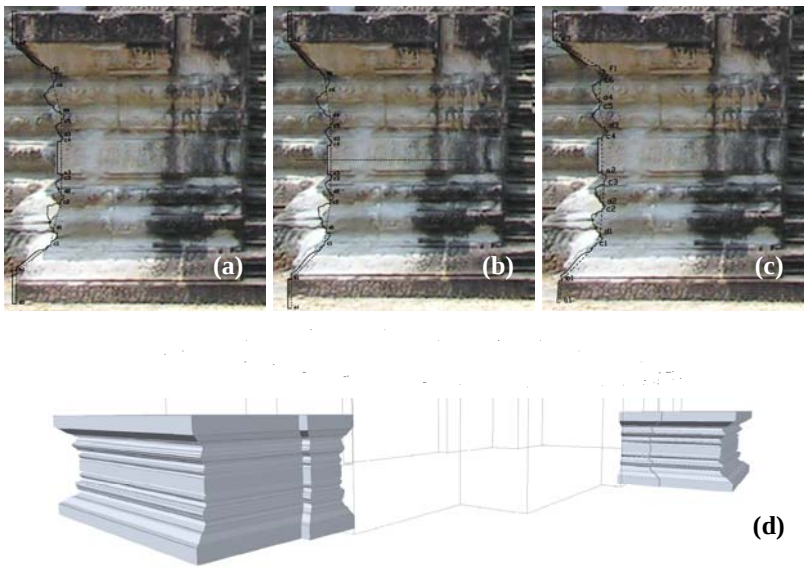


Figure 5.1.3
Comparison of the reconstructed profiles with the profile of the actual basement B1. (a) Using ATG with predefined module proportion settings. (b) Using ATG with adjustments to the predefined proportions. (c) Using ATD capturing tool. (d) the 3D model of B1 using profile of case (c).

Image (a) of figure 5.1.3 shows the results of using this tool. We observe that only a few modules of the profile match with those in the image including ${}_1c, {}_2a, {}_6c, f$ and ${}_3a$. With some manual adjustments to the properties of the modules we obtain a better result. In this case it matches ${}_1a, b, {}_1c, {}_3c, f$ and ${}_3a$, see image (b). However we get a much better results when using ATD capturing tool to capture each module manually as shown in image (c). The 3D model in image (d) is the result of using the profile (c) with the corresponding extrude path B1 shown in right image of figure 5.1.2.

ATG is not yet equipped with all functions for controlling the module proportions. At the moment it consists of a fixed U-shaped function for controlling the lengths of the modules. We might obtain better results if implementing other functions that we have presented in chapter 3. Another problem we meet here is that the central module ${}_2a$ of B1 is not positioning exactly on the central axis of the structure which we noticed by a horizontal dashed line on image (b) whereas this ATG tool always produces a profile of which the central module lies exactly in the middle of the profile.

Basement B3 is a more complex structured element. Its composition is $S_{B3} := (2a, b, 6d, e, f)$, and again we try using the ATG main tool to reconstruct its profile. A serious mismatch can be seen also in this case as shown on image (a) of figure 5.1.4. With some adjustments to the initial proportions we could match ${}_1a$, b, ${}_1c$, e, ${}_8c$, f and ${}_2a$, see image (b). The same as for reconstruction of B1, the best result can be obtained here when applying the capturing tool of ATD (image (c), (c1) and (c2)).

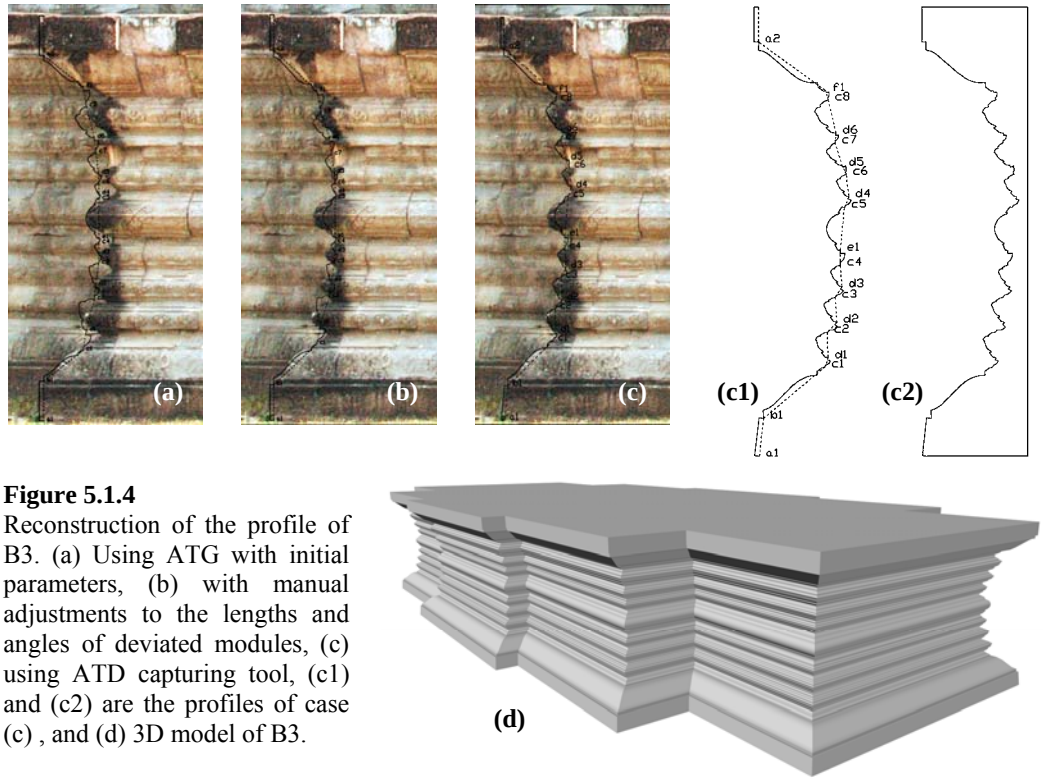


Figure 5.1.4
Reconstruction of the profile of B3. (a) Using ATG with initial parameters, (b) with manual adjustments to the lengths and angles of deviated modules, (c) using ATD capturing tool, (c1) and (c2) are the profiles of case (c) , and (d) 3D model of B3.

We observe that the module-based reconstruction method is ideal for the reconstruction of elements that is partly broken as the ${}_5d$ of B3 which is totally lost but we still could define its start- and end point, and thus could reconstruct its ideal state.

The remarkable property on basement B3 is the profile of module *e*. The central arc of the module profile is larger than that of ATG, and almost covers a semicircle. In order to reconstruct such extraordinary element profile we suggest using ATD detector which, of course, requires a suitable input image.

Extraordinary in terms of sequence structure is the wall base W1. We see that two convex modules, the ${}_1d$ and ${}_1b$, are adjacent modules which conflicts with the construction rule of ATG. With this available input image (b) as shown in figure 5.1.5, we have no other option then using the module capturing tool. Except for some minor artistic variations of forming the *d*-module profile (${}_1d$ and ${}_2d$), the result can be considered as perfectly matching with the origin. Temple element with such properties can only be reconstructed using either the ATD capturing tool or ATD detection tool.

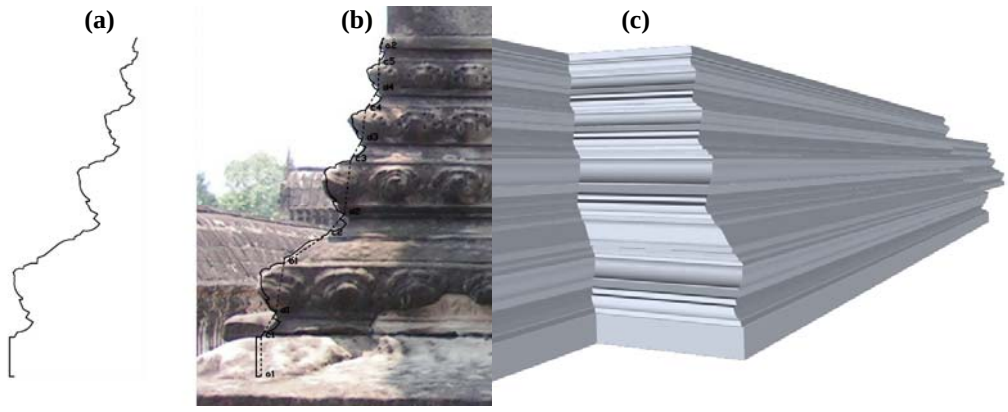


Figure 5.1.5
Reconstruction of wall base (W1) using ATD capturing tool. (a) Reconstructed profile of W1, (b) compare the result with input image, and (c) the 3D model of W1.

In contrast, the property of the wall capital W2 and the property of its input image for the 3D reconstruction allow us to have three options to find out the most appropriated reconstruction method. ATG tool provides a proper module sequence structure, however, also in this case it occurs some deviations which can be seen on

image (a1) of figure 5.1.6. The module capturing tool as expected provides is a better solution. In (b1) we observe a considerable mismatch of module $_3d$ that causes by its special function of what we would define as the *transitional module* from wall capital to the gutter or the other way round. The form of the lotus motif was probably depicted in order to adapt this transition, see the side cross section of the library in [Naf69].

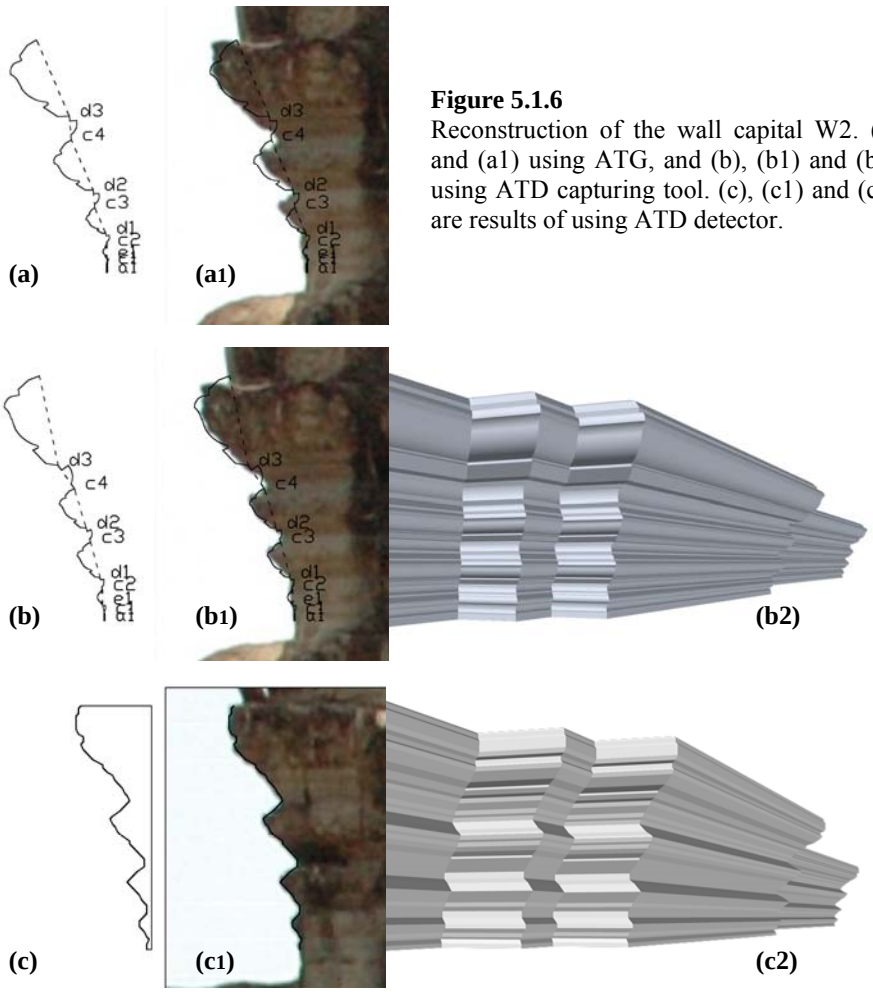


Figure 5.1.6
Reconstruction of the wall capital W2. (a) and (a1) using ATG, and (b), (b1) and (b2) using ATD capturing tool. (c), (c1) and (c2) are results of using ATD detector.

Image (c), (c1) and (c2) shows the results of implementing the detector tool, and we see the difference of the quality of both 3D model (b2) and (c2). We see that using the module-based reconstruction we obtain a clearer and smother geometric structure than that from the detector. The zigzag effect of model (c2) however, also caused by the low

resolution as well as the poor quality of the input image. We will see better results of using the detector a little bit later.

The column base of C1 consists of only three types of modules with the input $S_{column\ base} := (2a, 4c, 3d)$. As image (a) of figure 5.1.7 shows, ATG produces from these input data an appropriated module sequence: $S_{column\ base} = ({}_1a, {}_1c, {}_1d, {}_2c, {}_2d, {}_3c, {}_3d, {}_4c, {}_2a)$ but with serious mismatch with the photography of the element. A relatively acceptable result we obtain when using the module capturing tool. Image (b), (b1), (b2) and (c) shows the resulting 2D and 3D model respectively.

A crucial remark we must make at this point is that there is a certain simplification of the geometry of *d*-module profile. Observing the ideal geometry of ${}_1d$, the part that is bounded by the dash-lined square number 1 in image (b1) was not depicted on the module of the real column base. This simplification causes the mismatch in this reconstruction but also causes difficulties for identifying the joints of each module.

A fully illustrated form of *d* can be mostly found on large scale elements as well as elements that are directly faced to the eyes of the observer. An example for the latter case is the column capital. The simplified part of *d* on the column base can be clearly seen on the column capital, and is shown on image (e) of figure 5.1.8.

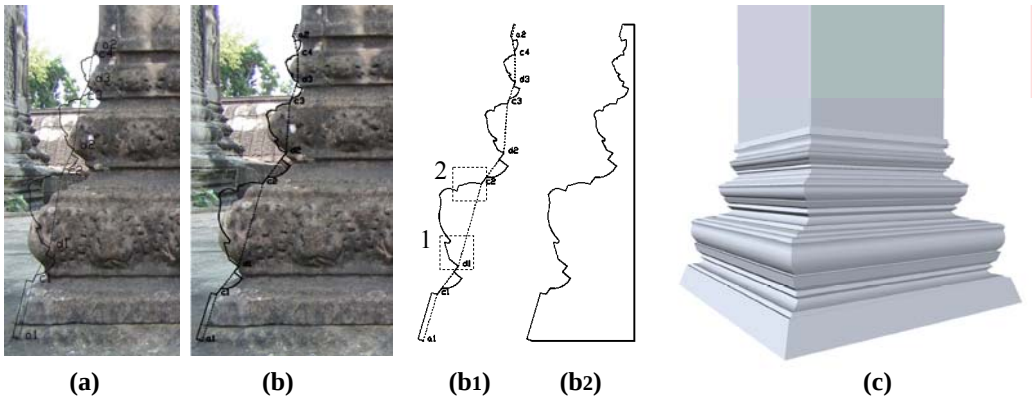


Figure 5.1.7
Reconstruction of the column base C1. (a) using ATG with adjustments of module proportions, (b) using ATD capturing tool, (b1) highlights the parts of potential simplification on *d*-module profile. (b2) are the results of (b), and (c) the 3D model of column base C1.

Let us demonstrate the 3D reconstruction of the column capital C1 by using the hand drawings made by the EFEO [Naf69] as input data. We discuss the reconstruction of this element in a close relationship with the other form of simplifying *d*-module profile that is marked by the dash-lined square number 2 on the image (b1) of figure 5.1.7.

From image (b) of figure 5.1.8 we observe that using the ATD module capturing tool it again, gives a much better match with the drawing if compared with case (a) which uses the ATG main tool. The reason for the deviation of module ${}_3d$ is similar to the case that we have discussed on the column base above, but here the upper part of *d*-module profile was left away. We see that the shape of ${}_3d$ appears alike that of *f*, which the draftsman of this drawing obviously discerns as the *f*. This can be seen clearly on (c) where we use the *f* instead of *d* which almost perfectly matches with the drawing.

To solve these problems we suggest two methods. The first is to use ATD profile detector with an available input photography of the element, and the second method is to adapt the simplification of *d*-module profile for the case of reconstructing column capital.

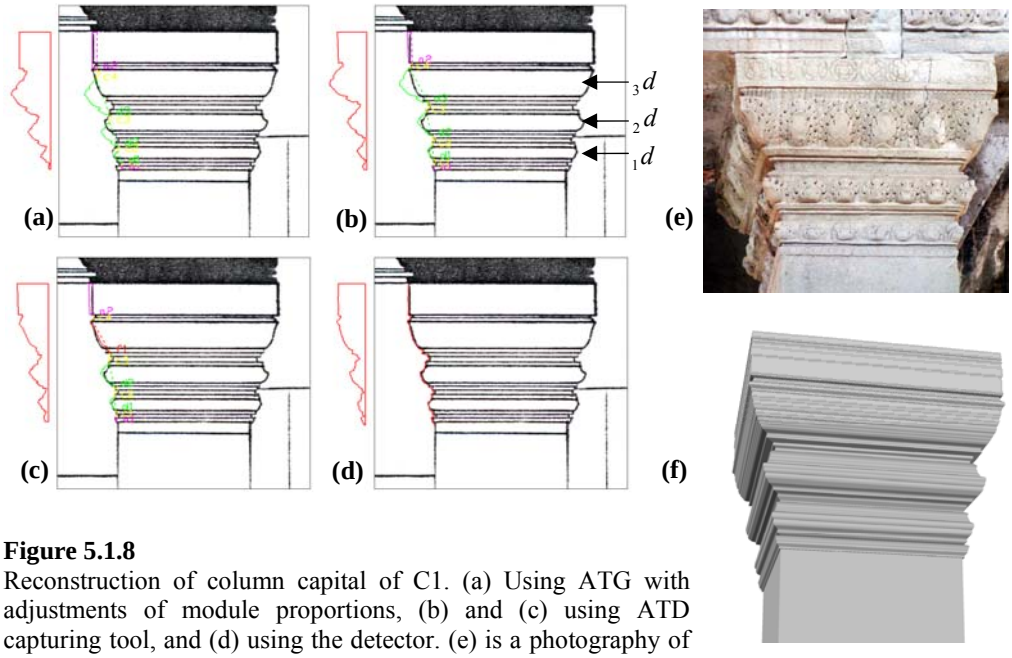


Figure 5.1.8
Reconstruction of column capital of C1. (a) Using ATG with adjustments of module proportions, (b) and (c) using ATD capturing tool, and (d) using the detector. (e) is a photography of the actual element, and (f) the 3D reconstruction using profile (d).

This means that we must improve ATG in such a way that it automatically simplifies the uppermost part *d*-module profile.

Image (d) and (f) show the results of the 2D and 3D reconstruction of C1 respectively using ATD detector to detect the profile from the drawing, and we can see that the 3D model is already almost identical to actual element in image (e).

The temple elements with highest structural complexity are the colonnettes and the window grids which do not belong to ATEs, and thus it is not possible to use the main tool of ATG to reconstruct them. But we can use the module capturing tool as we have demonstrated in figure 4.3.1, however, figure 4.3.2 also showed that there are some components that do not belong to ATE modules. It is therefore not always possible to apply this method. For the reason that such elements comprise of a large number of modules, practically it is suggested to use the profile detector to reconstruction them.

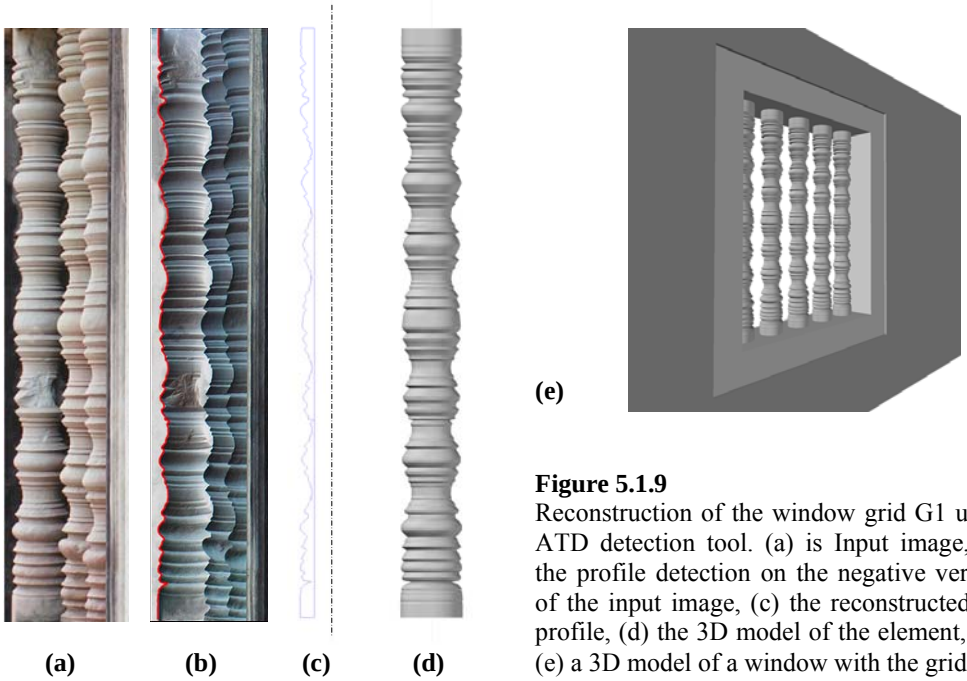


Figure 5.1.9
Reconstruction of the window grid G1 using ATD detection tool. (a) is Input image, (b) the profile detection on the negative version of the input image, (c) the reconstructed 2D profile, (d) the 3D model of the element, and (e) a 3D model of a window with the grids.

Figure 5.1.9 shows the reconstruction of the window grid G1 using a photography of the element as input image for the profile detection. We consider that, compared to the input image for reconstruction of W2 in figure 5.1.6, the resolution of this image is much

higher. As a result, we obtain a highly detailed and smooth 3D model which can be seen from image (d) and (e).

Now that all selected elements are reconstructed, we can assemble them together based on the positioning data we have obtained earlier in figure 5.1.1 image (e). We almost get a completed raw 3D model of the library as shown in figure 5.1.10. We see that the highly detailed 2D profiles of the elements produced by ATG tools finally give a high quality 3D model of the temple as a whole.

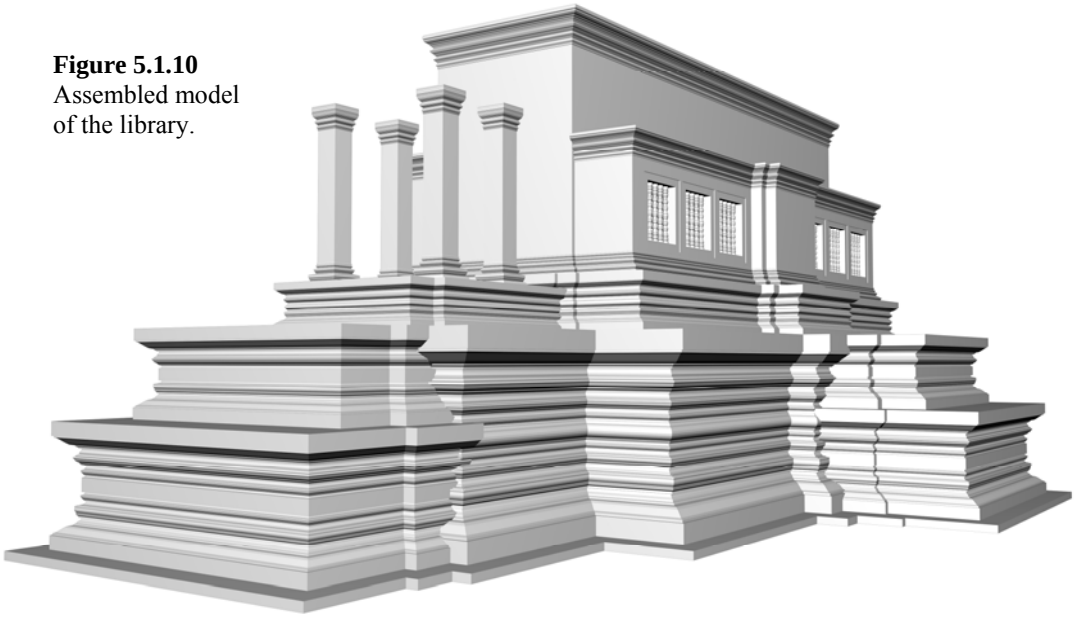


Figure 5.1.10
Assembled model of the library.

5.1.3 Virtual Reality and 3D Angkor Wat

A basic virtual reality rendering with texture mapping using only one photo of a typical Angkor temple sandstone layer grid applying to the complete 3D models of the library immensely enhance the realistic appearance of the temple, see figure 5.1.11.

Further results on using ATG to reconstruct the whole Angkor Wat temple complex is shown in figure 5.1.12, as well as in appendix C. All other results including images, animations, ATG and a test version of AutoCAD 2007, and an electronic version of this thesis are found in the attached DVD. All of them are copyrighted materials!

Figure 5.1.11
Both images are 3D Virtual Reality Rendering of the Library using Autodesk VIZ 2006 rendering tool based on the 3D model produced by ATG.

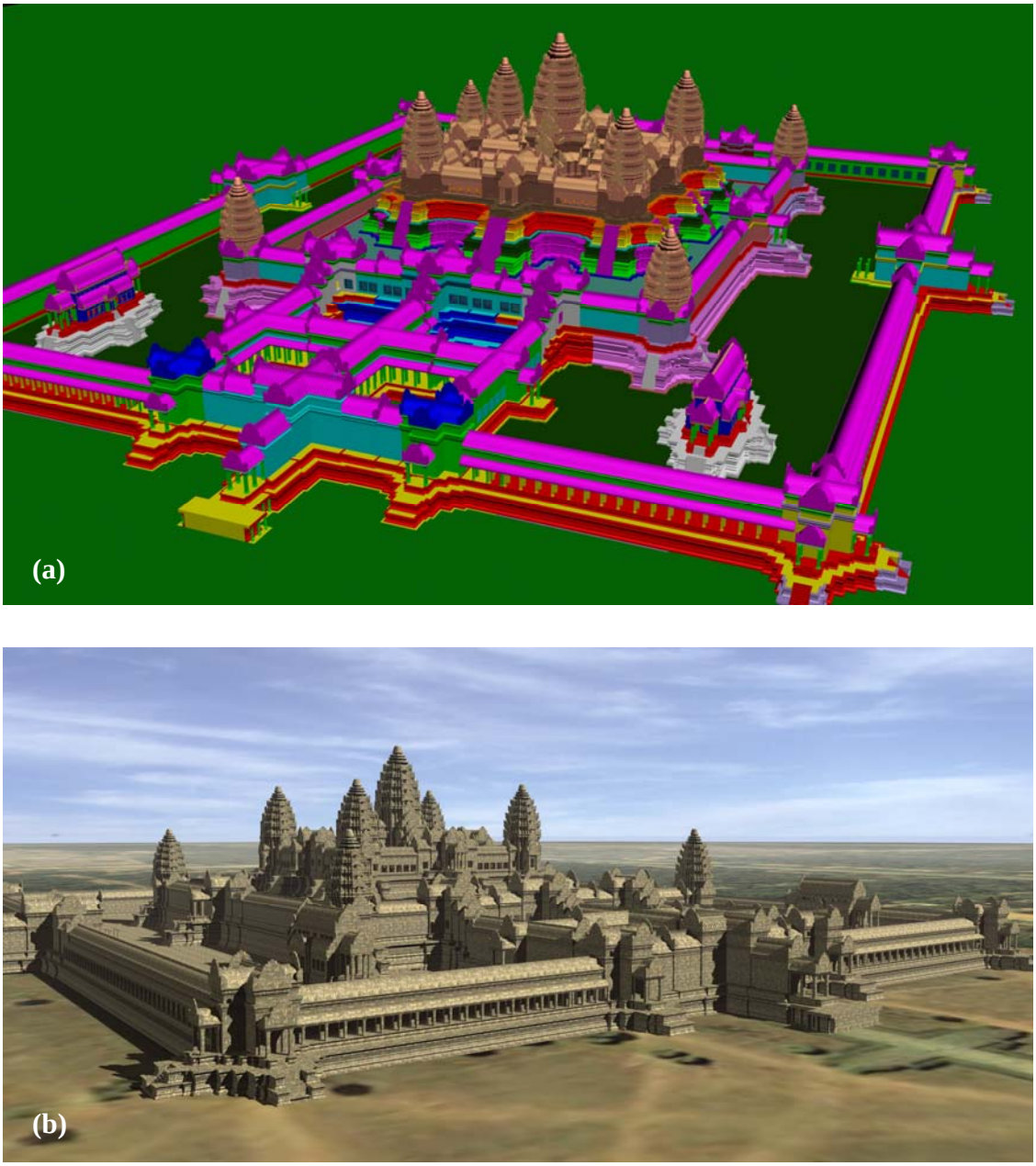


Figure 5.1.12
Computer reconstruction of Angkor Wat temple complex. (a) raw 3D model (uncompleted) of the temple complex of which elements are distinguished by colors, and (b) Virtual Reality Rendering of the temple using only one texture of a stone grid. The measurements for this 3D model are taken from drawings of [Naf69], EFEO.

We experienced that the reconstruction of complex structure such as Angkor Wat using highly detailed 3D models of elements can quickly leads to data overflow, and immensely slowdown the modeling process as well as the visualization performance. To avoid this problem we suggest reducing the level of details of small-scale elements and elements that are not supposed to face closely to the observer.

In order to obtain a more photo-realistic representation of the 3D model we must apply more advanced techniques in computer graphics. An important technique for the enhancement of the appearance of the fine art works especially the details of lotus motifs on the geometry of the modules is the advanced texture mapping technique. Each element should be applied with individual texture in such a way that the lotus motifs on the texture exactly match with the 3D geometry of the corresponding modules. Secondly, we must apply the bump mapping, to improve the bumpiness of the element surface thus bringing the carvings on the stone onto the front, see figure 5.1.13.

Figure 5.1.13
Computer reconstruction of the main basement of the central building of Angkor Wat temple.

In the following section we will present some samples of using ATG tools to reconstruct element profiles of temples that do not belong to Angkor Wat style temples. For our case study we select elements with special properties, and discuss them relative to the properties Angkor Wat styles elements as well as pointing out the most suitable reconstruction tool in each case to obtain the optimal results.

5.2 Reconstruction of Non-AWSTs

In this section we present some results of using ATG tools for reconstruction of temple elements for non-AWSTs. We distinguish these temples into two groups. The first Group consists of temples which have been constructed before Angkor Wat style period and the other after this period. For our reconstruction samples of the first group we select elements of the *Bakheng, Kravann* and the *Banteay Srey* temple, and from the second group the *Ta Prohm, Banteay Kdey,* and the great *Bayon* temple.

5.2.1 Temples before Angkor Wat Styles

At first sight we realize that the element components of earlier temple styles such as the Bakheng are the ATE modules very much alike. Without any doubt we could say that they provided architectural basis for later temple constructions thus also Angkor Wat style temples. The architectural structure of the uppermost terrace basement however, also consists of a component which does not belong to ATE modules; those corresponding to the ${}_2c$ and ${}_3c$. Nevertheless, they have the same architectural function as the *c*-module. We replace them with the *c,* and reconstruct the basement profile using the ATD module capturing tool as shown on image (a) of figure 5.2.1. The obtained structure $S_{basement} = ({}_1a, b, {}_1c, {}_1e, {}_2c, {}_2a, {}_3c, {}_1d, {}_4c, f, {}_3a)$ is not a symmetric module sequence structure as the basements structure of ATE. We observe that the *e*-module occur at the (common) position of the *d*-module.

Using the input image shown in this figure we can also apply the ATD profile detection tool to exactly reconstruct the basement profile as depicted on image (b). At this point we can summarize that the ATD profile detection tool is applicable for reconstructing all types of element profiles of any Khmer temples styles. Thus in the following discussion we will give no more samples on using it to reconstruct the element profiles.

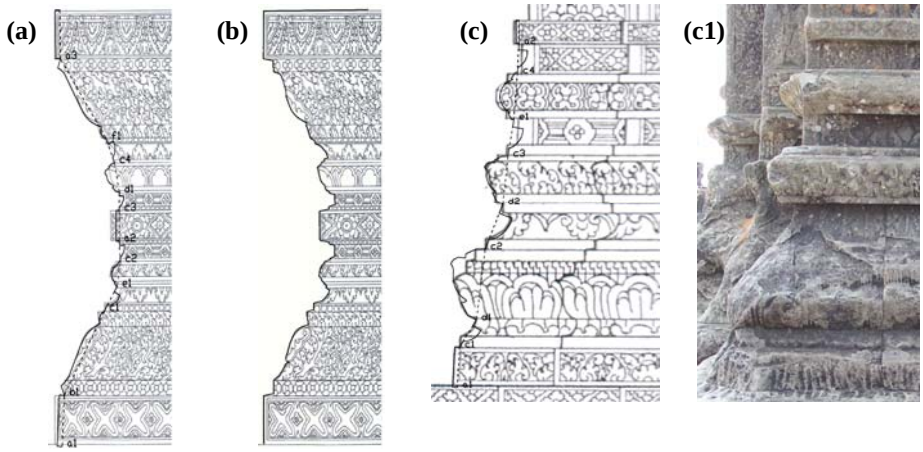


Figure 5.2.1
Reconstruction of two elements of Bakheng. (a) and (b) using module capturing and the ATD detector tool respectively to reconstruct the basement of the uppermost terrace. (c) capturing the modules of the wall base of the main tower. Drawings are done by EFEO.

The image (c) of figure 5.2.1 shows further samples on using module capturing tool to reconstruct the wall base of the central tower of Bakheng temple. The same as for the reconstruction of the basement profile in (a), a considerable mismatch here causes by the primitive form of *c*-module profile which are related to ${}_3c$ and ${}_4c$ of the reconstructed module structure. Observing ${}_2c$ from this image however, it clearly shows that the full geometry of *c*-module has already been used from this period (see chronology). But we also observe a very special combination of a simplified *d*-module with only a part of the ATE *c*-module profile which is the part that corresponds to $({}_1c, {}_1d)$.

Although both selected elements of Bakheng temple have properties of which reconstruction is beyond capabilities of ATG main tool, we still could use the ATD capturing tool and the detector to reconstruct their profiles with minor imprecision.

In about 921 AD, two decades after the reconstruction of Bakheng, the five towers brick stone temple of Kravann was constructed. On this temple we can clearer see the similarity between the components of their elements with ATE modules. But we also see some uncommon compositions such as the column capital of the central tower porch.

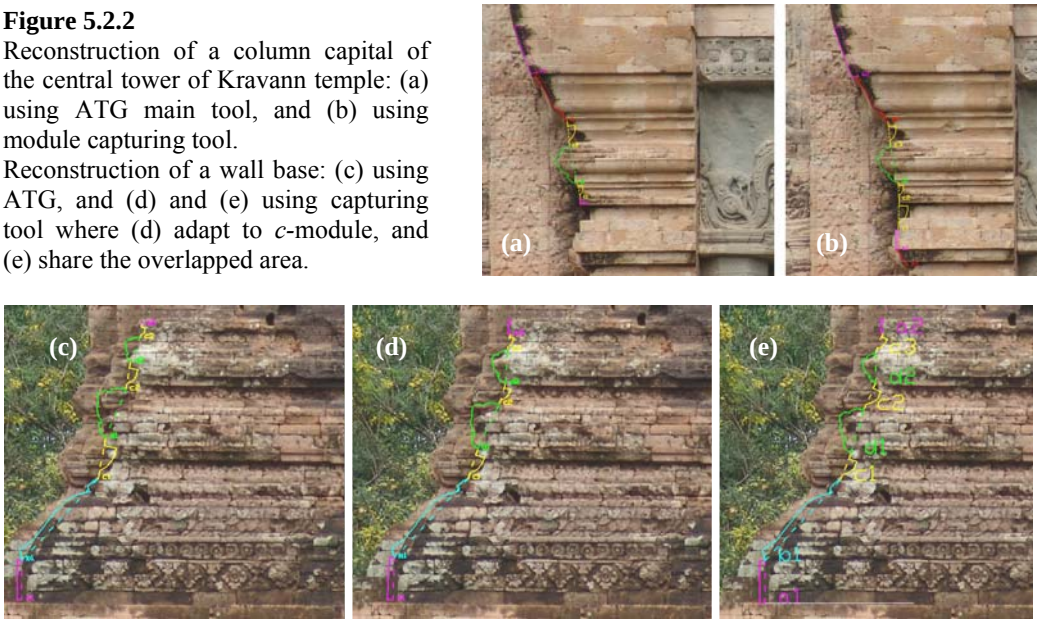


Figure 5.2.2
Reconstruction of a column capital of the central tower of Kravann temple: (a) using ATG main tool, and (b) using module capturing tool.
Reconstruction of a wall base: (c) using ATG, and (d) and (e) using capturing tool where (d) adapt to *c*-module, and (e) share the overlapped area.

In image (a) of figure 5.2.2 we show an ordinary ATE structure imbedded in the column capital structure of the Kravann temple. The whole structure of this column capital however, is a complex element. In order to reconstruct the whole structure with the most possible match, we need to improvise the module combination by forming a dual *c*-module structure below the *d*-module then connecting them with the last two ${}_1a$ and ${}_1f$, see image (b). We obtain $S_{column\ capital} = ({}_1f, {}_1a, {}_1c, {}_2c, {}_1d, {}_3c, {}_2f, {}_2a)$ of which architectural function of *c*-module is unclear, and uncommonly the *f* occurs twice.

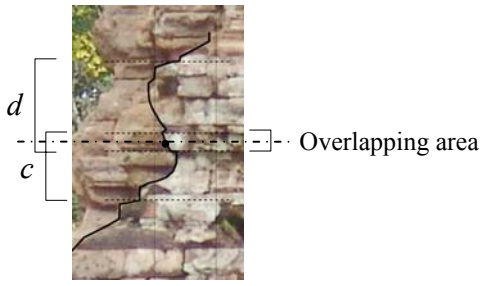


Figure 5.2.3
Adjacent modules *c* and *d* of the wall base shown in figure 5.2.2 which overlaps one another.

In contrast, the wall base of the central tower of Kravann is clearer structured, and satisfies the ATE structuring condition. It is therefore possible to use ATG to reconstruct the profile as shown on image (c). On this element we also observe simplified forms of

the *d*- and/or *c*-module. We use the expression "and/or" because a part of both profiles overlap each other. Image (d) of figure 5.2.2 shows the result of capturing the modules of the wall base in such a way that whole form of *c*-modules are preserved. Compared with the result as shown on image (e) we see a better match if we share the overlapped area in the middle to both modules as shown in figure 5.2.3.

The temple that is very well-known in grace and beauty of detailed fine art works is the Banteay Srey temple, constructed around 941 AD. Lotus motifs on their elements are very well depicted, even clearer than that of Angkor Wat style temples but their module profiles are still the earlier styles alike, the Kravann and Bakheng. We will see that the reconstruction results are also similar to the previous reconstruction cases however, and we begin two special characteristics of Banteay Srey.

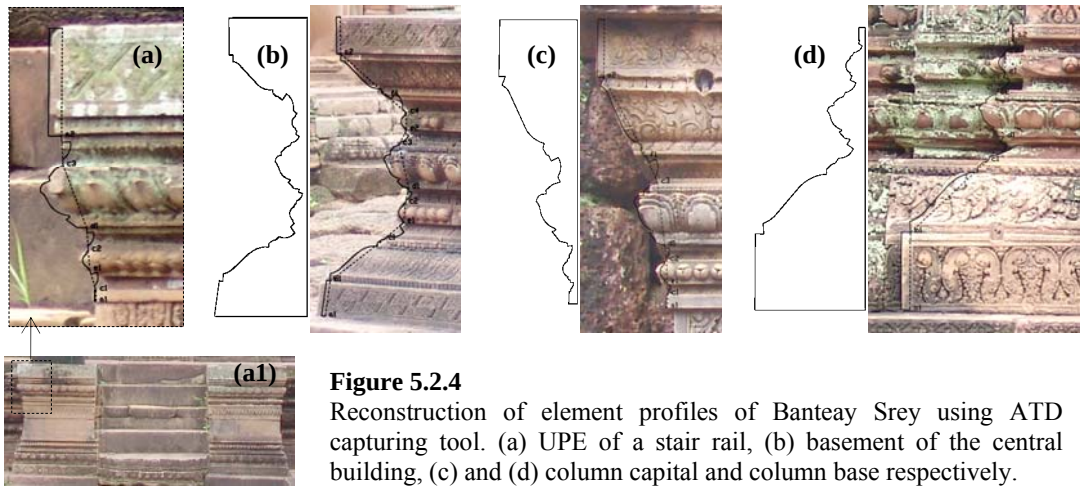


Figure 5.2.4
Reconstruction of element profiles of Banteay Srey using ATD capturing tool. (a) UPE of a stair rail, (b) basement of the central building, (c) and (d) column capital and column base respectively.

The first considerable element of Banteay Srey is the "banister", which for other temple styles normally is a part of a basement, and thus its profile is identical with the profile of that basement. In this case it is not. In relationship with ATE, we can consider it as a bounded wall with typical wall base and wall capital. Image (a) of figure 5.2.4 shows the reconstruction of its UPE using the module capturing tool (see also (a1)). We observe the minor mismatch causes by the simplified *d* profile and the primitive form *c*.

Another special module structure (element) if compared with Angkor Wat style temples, is the main basement of the central building (image (b)). Its module structure: $S_{basement} = ({}_1a, b, {}_1c, {}_1e, {}_2c, d, {}_3c, {}_2e, {}_4c, f, {}_2a)$ comprises of two of *e*-modules and one of *d*-module with the reversed architectural function of one another. We use the capturing tool to reconstruct the basement profile in (b).

The last two reconstruction samples of Banteay Srey elements are the column capital and column base shown on image (c) and (d) respectively. The module structures of both elements are typical ATE structures which allow us to use ATG main tool to reconstruct their profiles. We obtain better results however, when using the ATD module capturing tool as presented in that figure.

Conclusion: It is possible to use ATG tools to reconstruct the element profiles of earlier style temples before Angkor Wat style period, but in some cases with a limited matching precision if based on ATG modules and the predefined construction rule.

5.2.2 Temples after Angkor Wat Styles

Now let us discuss on using ATG tools to reconstruct elements of later temple styles. The three temples, The *Ta Prohm, Banteay Kdey* and *the Bayon*, belong to the group of temples constructed during the reign of the great king *Jayavarman VII*, 1181-1220 thus about three decades after Angkor Wat styles. We observe that the quality of carvings and lotus motifs presentation are a little bit lower than earlier temples however, during Jayavarman VII's period a large number of temples have been constructed as many as almost all other Khmer kings together, see chronology.

The elements of Ta Prohm temple (1186) are well structured with components which are very much ATE module structures alike. It is therefore possible to use ATG to reconstruct their profiles but here again the ATD module capturing tool provides better results as shown in figure 5.2.5. Ta Prohm might have elements with special module structure but our rough investigations could not confirm of such irregularity.

(a) (b) (c)

Figure 5.2.5
Reconstruction of elements of Ta Prohm temple. (a) wall base, (b) and (c) are respectively column base and column capital of a porch.

Similar to Ta Prohm can be observed from the Banteay Kdey temple. This temple was constructed around the same time with Ta Prohm, but is a little bit less complex structured. The artistic and structural properties of their elements are almost identical, and thus can be treated in the same way.

(a) (b) (c)

Figure 5.2.6
Reconstruction of elements of the Banteay Kdey temple using ATD module capturing tool. (a) and (b) are respectively a column base and column capital of a porch, and (c) a colonnette.

Our last samples are the reconstruction of element profiles of the great Bayon temple (1190), the second largest Khmer temple. Most module structures of its elements follow the construction rule of ATG. For this, we provide two examples on reconstructing a column base and a column capital as shown on image (a) and (b) of figure 5.2.7 respectively. We observe that the module sequence could be reconstructed by the

construction rule of ATG but the proportions of the modules are not the same as that of ATG. Case (a) can be reconstructed by ATG with less proportional adjustment however, case (b) requires too much adjustment, and thus the capturing tool is suggested instead.

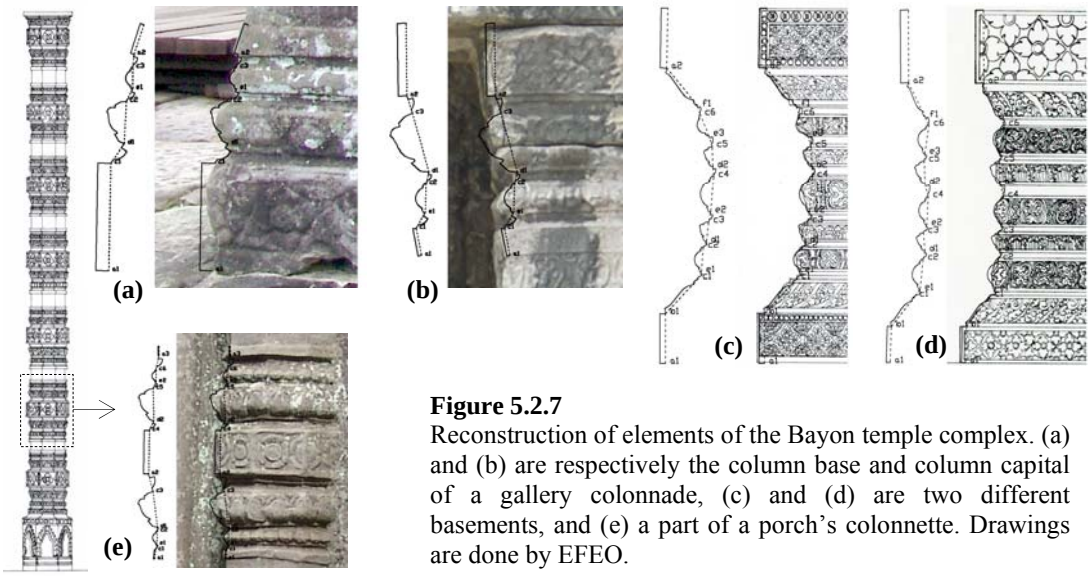


Figure 5.2.7
Reconstruction of elements of the Bayon temple complex. (a) and (b) are respectively the column base and column capital of a gallery colonnade, (c) and (d) are two different basements, and (e) a part of a porch's colonnette. Drawings are done by EFEO.

Image (c) (basement of northern staircase of Bayon complex) and (d) (main basement of its northern library) show extraordinary module sequences of two different basements of Bayon complex. There are basements of which module structures are identical to AWST basements, but these selected basements have a considerable structure: $S_{basement} = ({}_1a, b, {}_1c, {}_1e, {}_2c, {}_1d, {}_3c, {}_2e, {}_4c, {}_2d, {}_5c, e, {}_6c, f, {}_2a)$ with three *e*-modules of which two play the roll of *d*. Such structure could not be reconstructed by ATG. The results in (c) and (d) but also (a), (b) and (e), are provided by ATD capturing tool.

Another special factor that motivates us to show these two examples is the uncertain geometric form of the *e*-module on structure (c) if compared with (d). By looking closer at *e*-modules of (c) we realize that the carvings belong to a common *e*-module type however, its profiles rather fit to a *d*. In the reconstruction we did not identify them as the *d*-profiles because the predefined ideal profile of *d* in ATG would produce a 3D surface which would not match with the actual *e*-module texture. The

concept of combining a *d*-profile with an *e*-texture is without doubt a very interesting idea, and should be considered as part of the arts. Nevertheless, based on the predefined ideal modules we are constrained to improvise, and select the best possible way which in this case an *e*-module profile as case (d) also confirms.

The last reconstruction sample of case (e) shows the similarity of elements (quality of art works and module structure) of the temples that have been constructed during the reign of Jayavarman VII. We can compare (e) of figure 5.2.7 with (c) of figure 5.1.18. The overall structure of this colonnette comprise mainly of the cropped part of the whole colonnette that is shown on the left hand side of (e). Thus theoretically we need to reconstruct only a selected section, and copy it to place to the other parts.

Conclusion: There are not many different styles of temples after the period of Angkor Wat style temples though, it does not mean that fewer temples have been constructed. For the reason that most of them had been constructed within a short period of time in one reign of king, and probably designed by the same temple architects, the temples could be treated in the same way in terms of reconstruction. On the other hand, in most cases the architects seem to implement the construction rules imbedded in AWST which allow us to use almost all three tools of ATG to reconstruct them.

We can conclude that ATG main tool which is based on the predefined modules and the construction rules has the feasibility to provide proper module sequences but often with considerable mismatch with the input data. Thus it is not suitable for reconstruction of existing elements. Nevertheless, it is a useful tool and suitable for building new drafting of traditional Khmer temple design. The module capturing tool and the element profile detector of ATD on the other hand, can be applied to reconstruction almost any type of Khmer temple element with molding characteristics, and is suitable for reconstruction of existing objects.

Chapter 6

Conclusions and Outlook

Summary and Conclusions

The overall goal of this research is to provide computer-aided supports for preservation, restoration and reconstruction of the UNESCO world heritage temples of Angkor in Cambodia. With our "devotion to detail", and knowledge in Khmer arts and temple architecture we were able to generate a combined approach, together with the available state-of-the-art methodologies in applied mathematics, computer science and computer graphics, to develop a computer-aided design software tool called ATG which is specialized for highly detailed 3D reconstruction of Angkor Wat style temples.

With the results of this research we could confirm that one of the most important architectonic symbolisms of Angkor temples architecture is the sacred lotus flower which has been an essential iconography in Buddhism and Hinduism. We have discovered a construction rule which potentially had been an important part of the original construction rule that was developed by the ancient temple architects.

Description of the Method (Chapter 1 – 2)

First of all we have provided a rough overview of the history of Angkor temples relative to the problematic we are facing today. Most of the about one thousand years old temples of Angkor are still waiting for new technologies and methods for restoring the collapsed ruins whereas some are in danger of collapse. Some temples again are structural in a good

condition but their priceless carvings are shattering everyday which needed to be preserved and well documented as soon as possible.

Computer assisted design and methods are of great supports in this digital age however, the highly detailed structures and carvings of the temple elements cause vast difficulties for 3D computer reconstruction and modeling. Up-to-date known reconstructions are very limited quality 3D models, and moreover architectonic geometry is often improperly reconstructed.

With the knowledge in traditional temple architecture and construction philosophy as well as experience in 3D computer graphics and modeling we expect to be able to generate a user-friendly software tool of which reconstruction system is strictly based on the Khmer temple architecture, and thus could assists the reconstruction process.

The observations on Angkor temple architecture showed that there is an architectural pattern on the temple elements which are depicted as decorative moldings, and occur almost anywhere on the temple structure. The moldings comprise of particular composition of six different types of lotus motifs. The cross section of these lotus motifs are defined as the six modules of Angkor temple element structure.

Our investigations also showed that each module has its individual architectural function of which the proportions and the way they structured vary by the function the modules and the element type. The structuring pattern of the modules further allows us to setup a directed graph that represents the rule for structuring the six modules.

Based on these two major definitions we have developed the following:

- Six different names for the modules, *a, b, c, d, e* and *f,* and the data structures of their 2D module profiles geometry.
- Mathematical model in form of an adjacency matrix that represents the rule for structuring the modules.

Algorithmic Approaches and Software tool (Chapter 3 – 4)

In chapter 3 we present three different sets of algorithms each of which is for three different reconstruction methods of the ATG software tool:

- **ATG Main tool:** used for the reconstruction of 2D element profiles based on the predefined ideal module profiles and construction rule.
- **ATD module capturing tool:** used for the reconstruction of 2D element profiles based on the predefined ideal module profiles with the proportional and sequential properties captured from an input image.
- **ATD element profile detection tool:** used for the reconstruction of 2D element profiles from an input image by detecting the profile shape directly from the image.

The analytic data of the architectural functions of the modules that we obtained in the chapter 2 are used to setup the initial conditions in order to control the implementation of the construction rule thus producing a proper module sequence of the element structure. The conditions vary by groups of element types; LPE, UPE and elements with Non-alphabetic ordered module sequences, and are defined separately.

We have measured and analyzed the proportions of modules from nine selected photos of basements in order to find out the ideal functions of the length- and angle proportions of the module vectors. As a result, we found out that only the proportion of *c*- and *d*-modules require a function; the cosine function is suitable to apply for representation of their ideal length functions, whereas the angle proportion is a linear growth function.

The concept of ATD capturing tool is very simple. With the manual mouse clicks on the joints of each defined module of the element module structure on an input image, the system can reconstructs the current state of the element profile with ideal modules. The advantage of using this tool is also because it does not require a special input image.

However, the element structure must be visually well identifiable (by the user) as well as skills in recognition of module type are the essential prerequisites for using this tool.

For the reconstruction of elements which are still in a good condition, we further provided the ATD element profile detection tool. It is fully automated, and is applicable not only for reconstruction of elements of Angkor Wat style temples but any kind of elements (with molding) of any styles of Khmer temple. It is especially useful for reconstruction of very complex structured elements, and it provides options for controlling the levels of detail to optimize the geometry of the final 3D model.

Results and Experiments (Chapter 5)

ATG main tool still requires to be improved in its proportional control functions in order to use it for reconstruction of the existing elements. For the moment it is rather suitable for drafting new construction projects of traditional Khmer temples.

The module capturing tool is only partly automated, and suitable to reconstruct elements with limited structural complexity however, based on the reconstruction tests and results, it seems to be the best tool for reconstruction of element profiles which are eroded or partly lost.

ATD profile detection tool is a very user-friendly tool, and it is not only applicable for reconstruction of any Khmer temple element, but also other non-Khmer building of which the element geometry is depicted in form of moldings.

Outlook

We have seen in our reconstruction tests that there are mismatches between the profile of the actual elements and that generated by ATG main tool. There are few reasons that cause these mismatches:

- **Ideal proportion:** we need to develop advanced options for the adjustments of both proportional controls; the function of lengths proportion and angle proportion.

- **Advanced form of *c*-module:** we observe that in some cases, when the jut effect of an element is large; such as the basement of Banteay Srey temple (figure 5.1.16, image (b)), the shape of *c*-modules of the UPE is the vertical flipping of *c*-module of LPE. On the other hand, we have also seen a primitive form of *c*-modules on earlier style temples which can be considered as a horizontal flipping of the *a*-module. Thus we need to generate a family of *c*-module profile as well as new algorithms for placing them properly in the module sequence.
- **Advanced form of *d*-module:** we have mentioned two different form of the simplified *d*-modules geometry in figure 5.1.7. Both simplified *d*-module should be generated, and analyze their architectural function in order to setup an appropriated algorithm for the implementation.

We suggest developing a real-time matching of the predefined module profiles with a particular segment of a detected element profile from an input image. This can be done by fixing the end point of a selected module (selection is based on the general condition of definition 3.1.1 in chapter 3) at the initial point of detection on an image whereas the start point of the module is fixed to the last detected pixel of an input image.

As a result, we would obtain a sequence of ideal module instead of a chain of line segments of which module proportions are automatically detected from the image.

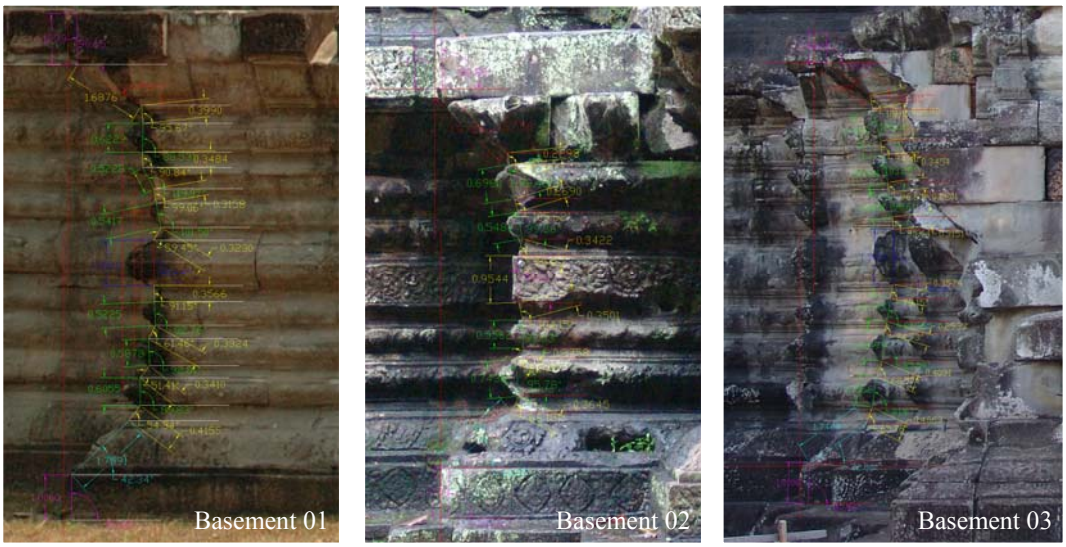
The ATD image viewer provides access to external hardware such as digital camera or video camera which gives us the possibility to develop a 3D scanner using the *depth from active triangulation* method which is one amongst many other of 3D imaging methods introduced by Jähne [Jän04]. It would be useful not only for 3D recording and reconstruction of element profiles with all the 3D lotus motifs but also for scanning and documenting the priceless carvings on the temple walls in 3D. Moreover, the obtained 3D data structure could be converted real-time in form of point clouds or triangles into AutoCAD. This 3D information could be very well combined with the 3D model of the temple generated by ATG tools.

Appendix A

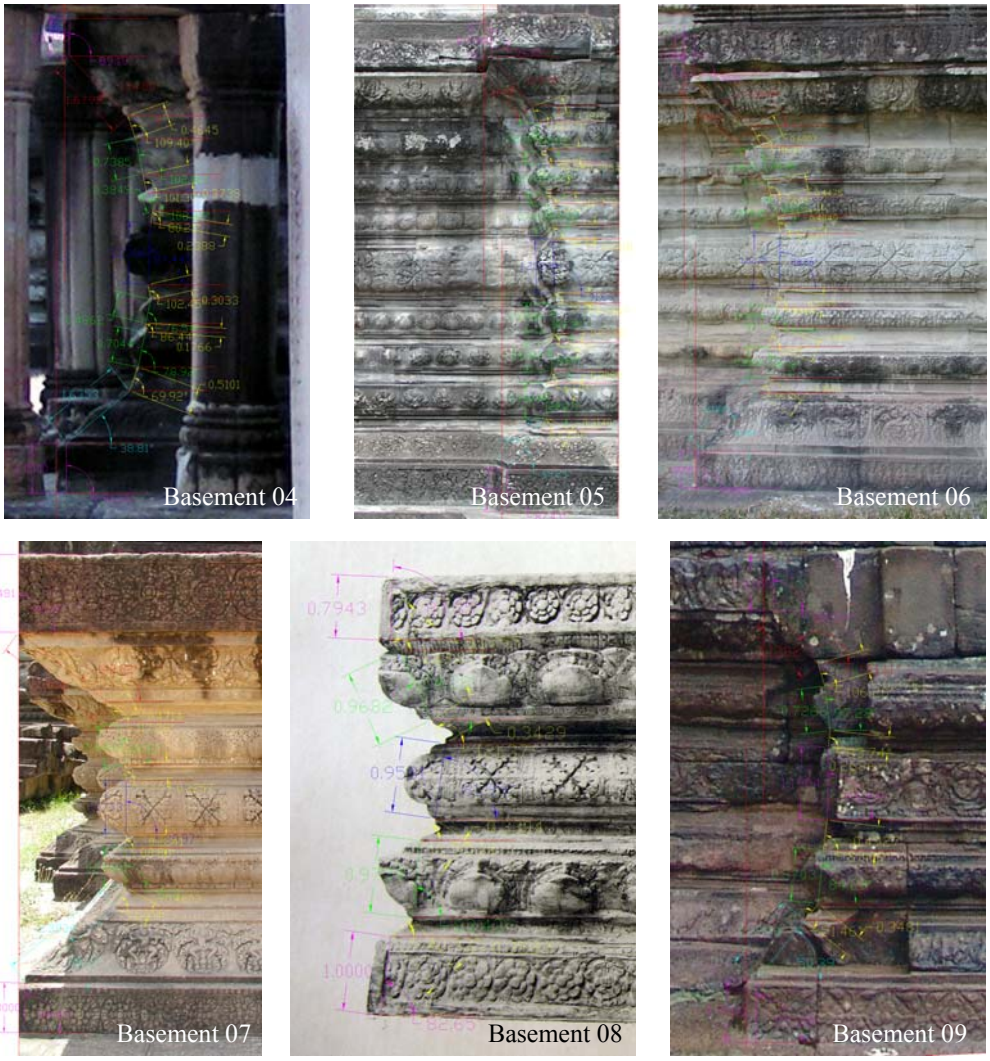
Figures and Tables

A.1. Measurements of Module Proportions from Images

In this appendix we present the materials and the way of measuring the module proportions of the nine selected basements used for the graphs in figure 3.1.10, 3.1.11, 3.1.14 and 3.1.15. The distinct colored dimensions correspond to the six different types of modules; *a* (magenta), *b* (cyan), *c* (yellow), *d* (green), *e* (blue) and *f*(red).



Basement 01; is the main basement of the southern library of Angkor Wat on the second level, Basement 02; the eastern main basement of the central building of Chao Say Tevda temple, and Basement 03 is the main basement of Angkor Wat central building.



Basement 04; basement of the terrace of Angkor Wat's western pavilion, Basement 05; southern main basement of Angkor Wat central building, Basement 06; main basement of a pavilion of Angkor Wat, Basement 07; basement of a stair case of Angkor Wat, Basement 08; uppermost level basement of a stair case of Angkor Wat, and Basement 09; main basement of Chao Say Tevda central building.

A.2. Symmetric Properties of the Angle Proportion of *c*- and *d*-modules

In section 3.1.3 we have analyzed the length and angle proportions of modules from the nine selected basements shown above. We observed a symmetric property of two

modules types; *c*- and *d*-modules from the module sequences. Such cases are shown Table A1 marked with a (*) symbol.

Basement 01					Basement 02				
index i	α_{c_i}	α_{d_i}	$\pi-\alpha_{c_{n_c+1-i}}$	$\pi-\alpha_{d_{n_d+1-i}}$	index	α_{c_i}	α_{d_i}	$\pi-\alpha_{c_{n_c+1-i}}$	$\pi-\alpha_{d_{n_d+1-i}}$
8	95.67					x			
7	90.84	88.53			6	95.36	x		
6	99.06	119.07			5	105.62	99.43		
5	59.45	101.5			4	76.65	99.36		
4	91.15	92.76	120.55	78.5	3	104.15	90.83	103.35 *	80.64
3	61.48	92.81	80.94	60.93	2	52.1	95.78	74.38	80.57 *
2	51.41	89.83	89.16	91.47 *	1	66.18	x	84.64	#VALUE!
1	54.34		84.33			x		#VALUE!	
Basement 03					Basement 04				
8	101.01					x			
7	94.4	106.13			6	109.4	x		
6	96.65	103.25			5	101.3	102.06		
5	95.44	93.61			4	80.21	108.45		
4	79.15	81.76	84.56 *	86.39 *	3	102.45	76.54	99.79 *	71.55 *
3	74.73	85.24	83.35	76.75	2	86.44	78.92	78.7	77.94 *
2	68.57	81.67	85.6	73.87	1	69.92	x	70.6 *	#VALUE!
1	65.76		78.99			x		#VALUE!	
Basement 05					Basement 06				
8	107.71	x				x			
7	95.59	98.57			6	116.41	x		
6	111.27	94.03			5	101.09	100.38		
5	90.44	99.25			4	97.83	110.41		
4	90.59	89.35	89.56 *	80.75	3	82.64	87.04	82.17 *	69.59
3	73.91	91.6	68.73 *	85.97	2	70.57	90.92	78.91	79.62
2	67.04	89.87	84.41	81.43	1	55.13	x	63.59	#VALUE!
1	63.1	x	72.29			x		#VALUE!	
Basement 07					Basement 08				
4	117.28	x			4	139.07	x		
3	93.64	100.51			3	121.37	114.28		
2	88.83	90.02	86.36 *	79.49	2	55.16	83.43	58.63 *	65.72
1	63.06	x	62.72 *	#VALUE!	1	40.93	x	40.93 *	#VALUE!
Basement 09									
4	106.26	x							
3	83.74	97.28							
2	83.08	84.29	96.26	82.72 *					
1	51.46	x	73.74	#VALUE!					

Table A1: List of measured angles of *c*- and *d*, and the symmetric property of angle proportion.

Appendix B

Calculation of the Module Proportion

B.1 Ideal Angle of $_1a$ and $_2a$ of UPE and LPE

Our investigation on thirteen LPEs and UPEs shows that the initiator $_1a$ of an UPE and the terminator $_2a$ of an LPE $\alpha_{_1a} = \pi - \alpha_{_2a} \cong \frac{5}{12}\pi$ with $_1a \in S_{UPE}$ and $_2a \in S_{LPE}$. Thus we define $\alpha_{_1a} = \pi - \alpha_{_2a} = \frac{5}{12}\pi$ as the ideal angles of the related elements.

B.2 Angle Proportion of LPE

The ideal angle proportion α_t of modules for $t = \{c,d\}$ and $\{a,e\} \in t(d)$ in LPE vary from $\alpha_{_it}$ to $\alpha_{_{i+1}t}$ within the interval $\left[\alpha_{_1t}, \frac{\pi}{2}\right]$. The difference $\Delta\alpha_t$ between $\alpha_{_it}$ and $\alpha_{_{i+1}t}$ is calculated based on (3.11) section 3.1.3:

$$\Delta\alpha_t = \frac{\alpha_{_{n_t}t} - \alpha_{_1t}}{n_t - 1} = \frac{\frac{\pi}{2} - \alpha_{_1t}}{n_t - 1}$$

The angle value $\alpha_{_it}$ at the index *i*:

$$\alpha_{it} = \alpha_{1t} + \Delta\alpha_t(i-1) = \alpha_{1t} + (i-1)\left(\frac{\frac{\pi}{2} - \alpha_{1t}}{n_t - 1}\right) = \frac{\alpha_{1t}(n_t - 1) + (i-1)\left(\frac{\pi}{2} - \alpha_{1t}\right)}{n_t - 1}$$

$$= \frac{\alpha_{1t} n_t - \cancel{\alpha_{1t}} + i\frac{\pi}{2} - i\alpha_{1t} - \frac{\pi}{2} + \cancel{\alpha_{1t}}}{n_t - 1}$$

$$\Rightarrow \alpha_{it} = \frac{\frac{\pi}{2}(i-1) + \alpha_{1t}(n_t - i)}{n_t - 1}.$$

Appendix C

Further Results

Photo of the Library of Angkor Wat

Reconstruction of the Library of Angkor Wat

Photo of Angkor Wat Complex

Reconstruction of Angkor Wat Complex

Reconstruction of Angkor Wat Complex

Reconstruction of Angkor Wat Complex

Reconstruction of a column capital of Angkor Wat temple

Reconstruction of a column capital of Banteay Srey temple

Chronology

Date	Empire	King	Capital	(T)emple / (B)aray	Notes
Pre-Angkor Period					
550-598	Chenla	Bhavavarman I	Bhavapura (Sombor Prey Kuk)	Phnom Da (T), in Angkor Borey; Koh Kuk Krieng (T), in Sambor	Son of Sriviravarman
600-610	-	Mahendravarman	Khon Kaen / Bhavapura		Name Citrasena, younger brother of Bhavavarman. Moved Capital from Khon Kaen to Bhavapura at Bhavavarman's death.
615-628	-	Isanavarman I	Isanapura	Sambor Prey Kuk (Temple group) 615	Sombor Prey Kuk, Capital of Chenla established in 616. Isanavarman was the Son of Mahendaravarman. Funan is completely demised in this period
628-657	-	Bhavavarman II		Prei Kneng (T)	
657-681	Upper Chenla & Lower Chenla	Jayavarman I	Purandarapura (maybe in the south of TDs Cambodia) Bhavapura (Rebuild and extend till Chantabori in TDs Thailand)	Prasat Andet (T) in Kompong Thom	14. June 657 consecration of 2 shrines, 1 in Battambang, 1 in Prey Veng
706-800		unknown	UC= Sombhupura LC=Vyadhyapura	Kompong Preah (T), Kompon Thom Trapeang Phong (T); Ak Yum (T)	
Angkor Period					
802-850	Khmer Empire	Jayavarman II	Mahendraparvata Mountain (Kulen Mountain)	Prasat Damrei Krap(T), Prasat Rup Arak(T), Prasat Neak Ta(T), Aram Rong Chen (T) in Kulen Mountain	802 Year of Independence from Indonesia, Beginning of Angkor period, King proclaim himself as Devaraja; established Hariharalaya, which later change to Indrapura by Indravarman
850-877	-	Jayavarman III			
877-889	-	Indravarman I	Indrapura (Roluos)	Indratataka (B); Preah Ko (T)879; Bakong (T)881;	Indratataka was build 5 days after his coronation.
889-910	-	Yasovarman I	Yasotarapura / Angkor	Lolei (T)893, in Roluos; Bakheng (T)905, in Yasotarapura; Phnom Krom (T)897-900; Phnom Bok (T)897-900; Baksei Chamkrong(T)907; Eastern Baray (B); Thma Bay Kream (T), 900-910 Prasat Bei (T), Preah Vihear(T), 900-910	
910-923	-	Hashavarman I	Angkor	Kravan (T), 921	Hashavarman I was the son of Yasovarman
923-928	-	Isanavarman II			Isanavarman II was a brother of Hashavarman
928-941	-	Jayavarman IV		Koh Ker (T);	
941-968	-	Rajendravarman I		Eastern Mebon (T)953; Bat Chum(T)953; Pre Rup(T)961; Banteay Srei(T)967;	at Eastern Mebon, Rajendravarman's divine Rajentreshvara was concentrated on Friday, 28. January 953 at 11am.
968-1000	-	Jayavarman V	Angkor	Takeo(T)975; Northern and Southern Khleang (T) 1000	Takeo is a state Temple

Date	Empire	King	Capital	(T)emple / (B)aray	Notes
1001-1002	-	Udayadityavarman I			
1002-1050	-	Suryavarman I	Angkor	Royal Palace, Phimeanakas(T); Western Baray (B), Western Mebon (T) 1002	Suryavarman (maternal line of Indravarman I)and Jayaviravarman were both consecrated as supreme king at the same time, and there is struggle for power Total temple: 2
1050-1070	-	Udayadityavarman II	Angkor	Baphuon(T) 1060	Baphuon was state temple Total temple: 1
1070-1080	-	Harshavarman III			
1080-1107	-	Jayavarman VI	unknown	Phimai(T) 1080; Wat Phu (T) 1080-1107	Enable an extremely peaceful reign. Phimai dedicated to Buddhism Total temple: 2
1113-1150	-	Suryavarman II	Angkor	Angkor Wat(T) 1140; Beng Mealea(T) 1050; Chao Say Tevoda(T) 1150; Thommanon(T) 1113, Wat Phu Baray (B) in southern Laos	Angkor Wat was state temple Total temple: 4
1150-1165	-	Yashovarman II	Angkor	Banteay Samrae(T) 1050; Preah Pithu Group(T)	Repair or modified Beng Mealea, banteay Samre and Bakong Total temple: 2
1170-1177	-	Tribhuvanadityavarman	Angkor		1165-1177 Ankgor was dominated by Cham king, Jaya-IndravarmanIV
1181-1220	-	Jayavarman VII	Angkor Thom	Banteay Kdey(T), Srah Srang(B) 1180-1200; Banteay Thom(T), Banteay Prei(T), Prasat Prei(T), Prasat Krol Ko(T) 1181-1200; Ta Prohm(T) 1186; Elephant Terrace(T), Ta Nei(T) 11d90, Wall and 4 Gopuras of Angkor Thom(T), Bayon(T) 1190-1210; Ta Som(T) 1190; Preah Khan(T), Neak Poan(T) 1191; Jayatataka (B) 1191; Prasat Suor Prat(T)	Defeated the Cham king Jaya-Indravarman IV Total temple: 14
1220-1244	-	Indravarman II			Repaired Banteay Kdey, Tasom and Tan Nei
Post-Angkor Period					
1243-1295	-	Jayavarman VIII		Mangalartha(T)	Repaired or modified Beng Mealea and Banteay Samre
1295-1307	-	Indravarman III		Preah Palilay(T);	Invasion by Siamese
1431	-	Ponhea Yat	Phnom Penh		Fall of Angkor, and end of stone temple construction

Bibliography

[AAA03] G. Artese, V. Achili, M. Abitrante, *Dynamic 3D Representation of Architecture as a Design Tool: The Old Town of Cosenza*, International Workshop WG V/4 & INTCOM III/V, Vision Techniques for Digital Architectural and Archeological Archives, Ancona, Italy, 2003.

[AG89] Ian O. Angell, Gareth H. Griffith, *Praktische Einführung in die Computer-Graphik, mit zahlreichen Programmbeispielen,* Carl Hanser Verlag München Wien, Gesamtherstellung Druck- und Verlagshaus Alois Erdl KG, Trostberg, Germany1989.

[Bax94] Gregory A. Baxes, *Digital Image Processing, Principles and Applications*, John Wiley & Sons, Inc., Hoboken, New Jersey, 1994.

[Bla05] I. Blanchard, *The Geopolitical Situation, The Formation of the Southeast Asian Mandalas (Polities) to ca 850 AD, Polity, Society and Economy in South-East Asia, prior to 1570*, Teaching Materials, University of Edinburgh, School of History & Classics, 2005

[BM76] J.A. Bondy and U.S.R. Murty, *Graph Theory, with Applications*, *Department of Combinatorics and Optimization, University of Waterloo, Ontario, Canada*, The Macmillan Press Ltd., 1976.

[Bun02] F. W. Bunce, *The Iconography of Architectural Plans – A Study of the Influence of Buddhism and Hinduism on Plans of South and Southeast Asia*, D.K. Printworld (P) Ltd. Regd. Office: "Sri Kunj", F-52, Bali Nagar, New Delhi – 110 015, 2002.

[Cha05] V. Chan, *Kbach, A Study of Khmer Ornament,* Reyum Publishing, First Edition, February 2005.

[Coe63] G. Coedes,, *Angkor, An Introduction,* English Edition, Oxford University Press, Ely House, London W. I, 1963.

[Cot01] Marion Cottingham, *Mastering, AutoCAD VBA,* SYBEX Inc., 1151 Marian Village Parkway, Alameda, CA 94501, 2001.

[Cun00] Olivier Cunin, *Production of New Graphics Documents Depicting the Architectural History of the Bayon Temple*, Fifth Symposium on the Bayon, Centre de Recherche en Architecture et Ingénierie, 2000.

[DGL+03] F. Dekeyser, F. Gasperd, L. Luca, M. Florenzano, X. Chen, P. Leray, *Cultural Heritage Recording with Laser Scanning, Computer Vision and Exploitation of Architectural Rules*, International Workshop WG V/4 & INTCOM III/V, Vision Techniques for Digital Architectural and Archeological Archives, Ancona, Italy, 2003.

[Die01] Reihard Diestel, *Graph Theory with Applications, Electronic Edition 2000*, Springer-Verlag New York 1997, 2001.

[EK06] D. Evans, M. Kummu, *Modeling Cultural and Natural Hydrology Using Rada Imaging at Angkor, Cambodia,* 2006

[Gla97] M. Glaize, *Angkor, A Guide to the Angkor Monument,* A translated from the 4th French edition, EFEO, 1997.

[Jac06] C. Jacques, *History of the Phnom Bakheng Monument,* Phnom Bakheng Workshop on Public Interpretation, CKS, Wat Damnak, Siem Reap, 2006..

[Jac99] C. Jacques, *Angkor,* Könemann Verlagsgesellschaft mbH, Cologne, 1999.

[Jäh02] Bernd Jähne, *Digitale Bildverarbeitung, 5., überarbeitet und erweiterte Auflage*, Springer-Verlag Berlin Heidelberg, 2002.

[Jäh04] Bernd Jähne, *Practical Handbook on Image Processing for Scientific and Technical Applications*, CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431, 2004.

[Lec14] A. Leclère, *Histoire Du CAMBODGE, Depuis Le 1er Siècle de Notre Ère,* Translated in to Khmer by Tep Mengkhean, Librairie Paul Ceuthner 13, Rue Jacon 13, Paris, 1914.

[Lev01] R. M. Levy, *Computer Reconstruction, Temple Site at Phimai, Thailand,* Faculty of Environmental Design, University of Calgary, 2001

[LGB03] G. L. Amato, G. G. Antonucci, B. Belnato, *The Three Dimensional Laser Scanner System: The New Frontier for Surveying. Case History: The Leaning Tower of Pisa (Italy), The Ancient Theatre of Taormina (Italy), The Prehistoric Sit of Nola (Naples-Italy)*, International Workshop WG V/4 & INTCOM III/V, Vision Techniques for Digital Architectural and Archeological Archives, Ancona, Italy, 2003.

[Män88] Martti Mäntylä, *An introduction to Solid Modeling*, Computer Science Press 1803 Research Boulevard Rockville, Maryland 20850, USA, 1988

[Naf69] Guy Nafilyan avec la collaboration de: Alex Turletti, Mey Than, Dy Proeung, Vong Von, *Ankgor Vat, Description Graphique du Temple,* ÈCOLE FRANCAISE D'EXTRÊME-ORIENT, Librairie Adrien-Maisonneuve, II, rue Saint-Sulpice, Paris 1969

[NR04] Takao Nishizeki, Md. Saidur Rahman, *Planar Graph Drawing, Lecture Notes Series on Computing - Vol. 12,* World Scientific Publishing Co. Pte. Ltd., Singapore, 2004.

[ON00] T. Ono, Ein Koproduktion von NHK Enterprises 21, Inc., *Angkor Wat, Das Geheimnis des Wasserreichs,* Bayerisches Fernsehen, 2000.

[Ort06] J. Ortner, *Angkor, Celestial temple of the Khmer Empire,* Abbeville press publishers New York, London, 2006.

[Pao03] Alberto Paoluzzi, *Geometric Programming for Computer-Aided Design*, John Wiley & Sons, Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2003.

[Pap01] Lothar Papula, *Mathematik für Ingenieure und Naturwissenschaftler Band 1,* Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 2001.

[Pra07] William K. Pratt, *Digital Image Processing, PIKS Scientific Inside,* John Wiley & Sons, Inc., Hoboken, New Jersey, 2007

[PS82] Christos H. Papadimitriou, Kenneth Steiglitz, *Combinatorial Optimization: Algorithms and Complexity,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

[Roo01] D. Rooney, *Angkor - an Introduction to the Temples,* Odyssey Publications Ltd., 2001.

[Sch02] Peter Schreiner, *Im Mondschein öffnet sich der Lotus. Der Hinduismus,* Deutscher Taschenbuch Verlag, München, 2002.

[SSRS06] Till Sonnemann, Martin Sourbier, Fabio Remondino and Gerhard Schrotter, *Reality-based 3D Modeling of the Angkorian Temple Using Aerial Images,* Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland, 2006

[Sta94] M. Standen, *Passage through Angkor,* Mark Standen Publishing Company Ltd., 43/364 Moo Bahn Amarinnivet 1, Bangkhen, Bangkok 10220, Thailand, 1994.

[Sti02] H. Stierlin, *Hinduistisches Indien, Temple und Heiligtürmer von Khajuraho bis Madurai,* Taschen GmbH, Hohenzollernring 53, D-50672 Köln, 2002.

[Sti79] H. Stierlin, *Die Welt der Temple und Pagoden*, *Kunst und kultur Südostasiens under dem Einfluß Angkors,* Gondrom Verlag Bazreuth, 1979.

[Sti82] H. Stierlin, *Angkor, Architektur der Welt,* Benedikt Taschen Verlag GmbH, Germany, 1982.

[Sut04] Joe Sutphin, *AutoCAD 2004 VBA: A Programmer's Reference,* Apress, 2560 Ninth Street, Suite 219, Berkeley, CA 94710, 2004.

[Tau03] P. Taus, *Virtual Reality in the Study of Archeological Sites: A Data-base for Funum Fortunae*, International Workshop WG V/4 & INTCOM III/V, Vision Techniques for Digital Architectural and Archeological Archives, Ancona, Italy, 2003.

[YTSN03] O. Yamada, Y. Takase, I. Shimoda, T. Nakagawa, *Significance of Digital Reconstruction of Historic Buildings Using 3D Laser Scanner, Case Study: Prasat Suor Prat N1 Tower, Angkor, Cambodia,* International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, Natural Resources Canada, Great Britain, 2003.