Inaugural-Dissertation
zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität Heidelberg

vorgelegt von
Dipl. Math. Anja Schäfer
geb. in Heidelberg

Tag der mündlichen Prüfung: 12.01.2015

An optimizational approach for an algorithmic reassembly of fragmented objects

Advisors: Prof. Dr. Bock
Prof. Dr. Leitte

Danksagung

Mein besonderer Dank gilt an erster Stelle Prof. Dr. Hans Georg Bock für das interessante Thema, die Möglichkeit, nach Kambodscha zu gehen, die Konferenzen, zu denen ich durfte und die mir gewährten Freiräume bei der Untersuchung des Puzzles. Prof. Dr. Heike Leitte möchte ich ebenfalls besonders danken, die mit Geduld meine Gedanken und Ideen in die richtigen Bahnen lenkte und deren Tür immer für Fragen und Diskussionen offen stand. Beiden meinen Betreuern möchte ich meinen Dank aussprechen für konstruktive Kritik und anregende Gesprächsrunden. Es hat mir sehr viel Spaß gemacht.

Den Arbeitsgruppen *Computergraphik und Visualisierung (CoVis)* und *Visualisierung und numerische Geometrie (vNGG)* möchte ich für die interessanten Diskussionen beim Mittagessen und "in Kamerun" danken, ebenso wie für die Diskussionen, Anregungen und Tipps rund um meine Arbeit. Besonders danke ich in diesem Zusammenhang auch Bastian, der (fast) alle meine Fragen zu beantworten wusste. Danke auch für die spannenden morgendlichen Kaffeediskussionen. Mein Dank geht ebenfalls an meinen Forschungsstudenten Fabian, der mich tatkräftig unterstützt hat.

Den Sekretariaten des IWR und der HGS MathComp möchte ich für die Unterstützung im Verwaltungsdschungel danken. Ohne sie wäre ich an den bürokratischen Stolpersteinen gescheitert.

Danke an Prof. Dr. Bock und die HGS für die finanzielle Unterstützung, sowie John Sanday und dem Global Heritage Fund für die Unterstützung in Banteay Chhmar.

Mein größter Dank geht an Julia und an Pubsch. Erstere stand mit geduldigen Fragen, Fachwissen sowie Rat und Tat immer zu Seite und beide halfen mir stets, auf den Boden der Tatsachen zurückzukehren. Ohne euch und eure Unterstützung hätte ich diese Arbeit nicht beendet. Ihr wart da, wenn nichts mehr ging und die Lage (besonders) aussichtslos erschien.

Darüberhinaus möchte ich mich bei all denjenigen bedanken, die sich die Mühe gemacht haben, Teile meiner Doktorarbeit Korrektur zu lesen, vor allem Nina, die den größten Teil übernommen hat.

Auch möchte ich mich bei meiner Familie bedanken. Sie ermöglichte mir mein Studium und legten den Grundstein für mein mathematisch-naturwissenschaftliches sowie kulturelles Interesse.

Schließlich danke ich auch der Fakultät, sowie allen, die mich bei der Erstellung meiner Doktorarbeit unterstützt haben, auch jenen, die ich vergessen haben könnte, hier namentlich zu nennen.

Abstract

In Cambodia close to the Thai border, lies the Angkor-style temple of Banteay Chhmar. Like all nearly forgotten temples in remote places, it crumbles under the ages. By today most of it is only a heap of stones. Manually reconstructing these temples is both complex and challenging: The conservation team is confronted with a pile of stones, the original position of which is generally unkown. This reassembly task resembles a large-scale 3D puzzle. Usually, it is resolved by a team of specialists who analyze each stone, using their experience and knowledge of Khmer culture. Possible solutions are tried and retried and the stones are placed in different locations until the correct one is found. The major drawbacks of this technique are: First, since the stones are moved continuously they are further damaged, second, there is a threat to the safety of the workers due to handling very heavy weights, and third because of the high complexity and labour-intensity of the work it takes several months up to several years to solve even a small part of the puzzle.

These risks and conditions motivated the development of a virtual approach to reassemble the stones, as computer algorithms are theoretically capable of enumerating all potential solutions in less time, thereby drastically reducing the amount of work required for handling the stones. Furthermore the virtual approach has the potential to reduce the on-site costs of in-situ analysis. The basis for this virtual puzzle algorithm are high-resolution 3D models of more than one hundred stones. The stones can be viewed as polytopes with approximately cuboidal form although some of them contain additional indentations. Exploiting these and related geometric features and using a priori knowledge of the orientation of each stone speeds up the process of matching the stones.

The aim of the current thesis is to solve this complex large-scale virtual 3D puzzle. In order to achieve this, a general workflow is developed which involves 1) to simplify the high-resolution models to their most characteristic features, 2) apply an advanced similarity analysis and 3) to match best combinations as well as 4) validate the results.

The simplification step is necessary to be able to quickly match potential side-surfaces. It introduces the new concept of a minimal volume box (MVB) designed to closely and storage efficiently resemble Khmer stones. Additionally, this reduced edge-based model is used to segment the high-resolution data according to each side-surface. The second step presents a novel technique allowing to conduct a similarity analysis of virtual temple stones. It is based on several geometric distance functions which determine the relatedness of a potential match and is capable of sorting out unlikely ones. The third step employs graph theoretical methods to combine the similarity values into a correct solution of this large-scale 3D puzzle. The validation demonstrates the high quality and robustness of this newly constructed puzzle workflow.

The workflow this thesis presents virtually puzzles digitized stones of fallen straight Khmer temple walls. It is able to virtually and correctly reasemble up to 42 digitized stones requiring a minimum of user-interaction.

Zusammenfassung

In Kambodscha nahe der thailändischen Grenze befindet sich die Tempelanlage von Banteay Chhmar, erbaut im Stile des besser bekannten Angkor Wat. Wie die meisten fast vergessenen Tempel zerfällt sie zusehends und ist heutzutage kaum mehr als ein großer Haufen Steine. Ein Wiederaufbau dieser zerfallenen Strukturen von Hand ist sehr komplex und fordernd: Ein Team aus Konservatoren und Archäologen steht vor einem Trümmerhaufen, bei dem die ursprüngliche Position der einzelnen Teile meist unbekannt ist. Generell wird diese Art von überdimensioniertem Puzzle gelöst, indem ein Team von Spezialisten jeden einzelnen Stein mit Hilfe von Erfahrung und Wissen über die Kultur der Khmer analysiert. Mögliche Lösungen werden solange ausprobiert, bis die richtige Zusammensetzung der Steine gefunden wurde. Die größten Nachteile dieser Methode bestehen darin, dass die Steine bei jeder Bewegung weiter beschädigt werden und die Sicherheit der Arbeiter durch das Hantieren mit schweren Gewichten gefährdet ist. Darüber hinaus dauert diese Art des Wiederaufbaus aufgrund ihrer hohen Komplexität und Arbeitsintensität mehrere Monate oder Jahre.

Diese Bedingungen motivierten die Entwicklung eines virtuellen Ansatzes, um den Wiederaufbau zu unterstützen, da computerbasierte Algorithmen das Potential haben, die Lösung schneller zu finden und gleichzeitig den notwendigen Arbeitsaufwand reduzieren können. Desweiteren können die Kosten vor Ort gesenkt werden. Die Basis dieses virtuellen Puzzles sind hochauflösende 3D Modelle von mehr als hundert Steinen. Diese Steine können als annähernd quaderförmige Polyeder aufgefasst werden, häufig mit einer zusätzlichen Einkerbung versehen. Das Ausnutzen dieser Eigenschaft in Kombination mit Vorwissen über die Orientierung eines jeden Steins führt zu einem schnelleren Arbeitsablauf beim Zusammenfügen der Steine.

Das Ziel der vorliegenden Arbeit ist es dieses virtuelle 3D Puzzle zu lösen. Die zugrunde liegende Idee dabei ist 1) die hochauflösenden Modelle auf ihre charakteristischsten Eigenschaften zu vereinfachen, 2) eine ausgeklügelte Ähnlichkeitsanalyse anzuwenden und 3) die besten Kombinationen zusammenzusetzen sowie 4) die Ergebnisse zu validieren.

Der Vereinfachungsschritt ist notwendig, um mögliche Lösungen schnell zu testen und zusammenzusetzen. Hierbei kommt eine neue Art der Repräsentation von Khmer Tempelsteinen zum Einsatz, die *minimale Volumen Box (MVB)*. Sie wurde entwickelt, um die Steine speichereffizient und trotzdem genau wiederzugeben und um die hochauflösenden Daten entsprechend der Seitenflächen zu segmentieren. Im zweiten Schritt wird eine neue Technik vorgestellt, die es ermöglicht, eine Ähnlichkeitsanalyse der Steine durchzuführen. Sie basiert auf verschiedenen geometrischen Abstandsmaßen, die Aussagen über die Zusammengehörigkeit eines möglichen Paares treffen. Die besondere Stärke dieser Analyse ist, dass unpassende Kombinationen aussortiert werden. Im dritten Schritt wird mit Hilfe von graphentheoretischen Methoden die korrekte Lösung des Puzzles gefunden. Abschließend wird die hohe Qualität und Stabiliät dieses neuartigen Khmer 3D Puzzles präsentiert.

Der Arbeitsablauf, den diese Arbeit präsentiert, ist in der Lage, digitalisierte Steine von zerfallenen geraden Mauern von Khmer Tempeln zu puzzlen. Es ist nun möglich, bis zu 42 virtuelle Steine mit einem Minimum an Benutzeraufwand wieder zusammenzusetzen.

Contents

Chapter 1

Introduction

> *For the enthusiast of Khmer art, just the name* Banteay Chhmar *will evoke lively interest, mingled with passion and wonder.*[1]

1.1 Motivation

All over the world, vast numbers of ancient temples are crumbling under their age, forgotten in remote places. Most of them are hardly more than a heap of stones by today. Archaeologists are fighting against time and decay in order to preserve and restore those fallen monuments. The evolution of computers, acquisition techniques and storage capacities opens up a whole new world of possibilities. An object can now not only be studied in-situ, where further constraints as for example accessibility or fragility have to be met or back in the office, using photographs and drawings, but also on a computer at any place and any time. Additionally, three-dimensional virtual models of an object help gain new insights without touching and thereby further deteriorating the object.

In the area of cultural heritage the vast majority of objects such as cuneiform tablets, vessels, potsherds or temples comes from architecture and ceramics as they are the most common findings in archaeology. Unfortunately, such man-made objects erode over time, break apart and are scarcely found unfragmented during excavations. This leads to the

[1] Cunin, O. and Baku, S.; The Face Towers at Banteay Chhmar; Tokyo 2005; p. 106

longsome and intensive work of finding the correct placement for each piece. The reason such time-consuming labor is conducted lies in the promising potential of learning something about the studied culture. Findings usually contain ornamental paintings, decorations, bas-reliefs or inscriptions and a complete object helps in reading the inscription or interpret the ornamental picture. Yet, determining the correct reassembly of broken artifacts is made difficult by many facts, e.g. the numerous objects and object parts that need to be compared to each other or that it is unknown whether the object is complete or how the final result will look like or the existence of matchings that are too small for the human vision. Consequently, reconstruction systems bear the potential to: save time, make the result more exact, gain deeper insights, find new combinations and probably complete objects whose fragments have been acquired at different times and different places.

The present dissertation deals with a partial reconstruction of a large but nearly forgotten temple site in northwestern Cambodia, called *Banteay Chhmar*. Due to looting and the ingression of plants only about 20% of the original structure are still standing. A real-world reassembly would be desirable to deepen the understanding and widen the knowledge about Khmer culture and history, but is not feasible because of several reasons such as:

- Location
  The accessibility of the site is complicated due to the fact that the temple is located in a sparsely settled area, with only unpaved roads and a broad belt of landmines – remainings from the war.

- Climate
  "In the summer, there is no game and torrid heat; in winter, the area is subjected to violent storms deflected by the mountains. This is the most desolate place in Cambodia."[2]

- Temple stones
  A manual reassembly would require several men to move the temple stones, where stone each can weigh between 100kg up to 1000kg making the handling difficult. This means every movement of a stone is able to increase its deterioration as well as endangers the workers involved.

[2]Groslier, George; Une merveilleuse cité khmère; L'Illustration magazine; Paris 1937; no. 4909; p. 352

1.2 Goals and Requirements

A computer-based reconstruction of the Banteay Chhmar temple side can be seen as a large-scale 3D puzzle where each stone represents one of its pieces and the (complete) reassembly is the solution. According to the Oxford English Dictionary *"to puzzle something out" means"* solve or understand something by thinking hard" and a *"(jigsaw) puzzle"* is *"a mystery that can only be resolved by assembling various pieces of information."*[3] Meriam Webster defines it as *"a question or problem that requires thought, skill, or cleverness to be answered or solved; something or someone that is difficult to understand."*[4] This virtual task has the potential to be much faster than the real one, reduce further deterioration of the temple stones and enhance the workers' security. In contrast to classical 2D jigsaw puzzles, which contain readily identifiable structures that enable an algorithm to easily find corresponding parts and uniquely match the pieces, broken fragments from man-made objects have no easy-to-identify corners or boundaries. The beginning or end of such structures is difficult to determine as well as each curve, corner or boundary *"may match equally well with numerous similar boundaries from other fragments."*[5] This is therefore a different approach thus the scope of this thesis is not only to discuss 3D puzzles in general thereby examining prerequisites that need to be met but also an interdisciplinary approach – a study of cultural features and the employment of knowledge from mathematics and computer science aims at developing an autonomous algorithm that is able to solve the puzzle. Main aspects of this algorithm are 1) the formulation of a robust distance measure for discrete surface models, 2) the development of optimization methods to deal with this class of discrete optimization programs as well as 3) the use of a graphical user interface to make the underlying complex algorithms manageable for the researcher in the field. To this end high-resolution 3D models of stones are needed as well as a map of their original placement within the temple. Only with a deep understanding of temple building and expertness in layout the proposed solutions of the algorithm can be validated and the correct reassembly be determined.

1.3 Workflow Overview

Before solving this digital 3D puzzle, the stones need to be acquired and digitized first. Digital acquisition equipment (also known as 3D scanner) is nowadays capable of cap-

[3] http://www.oxforddictionaries.com/definition/english/puzzle, accessed 27.05.2014

[4] http://www.learnersdictionary.com/definition/puzzle, accessed 27.05.2014

[5] A.R. Willis and D.B. Cooper, Computational reconstruction of ancient artifacts, IEEE Signal Processing Magazine 25 (2008), no. 4, p.67

turing objects at a micrometer resolution. The resulting output is a triangulated surface mesh model that nearly flawlessly reproduces the stone in all its details. The remaining challenge ist therefore that of reassembling the individual 3D stone models. To this end, this thesis proposes a five step *puzzle workflow* that is qualified to solve this task.

The first step is to simplify the virtual stone models to wireframe box models. This is done due to the fact that the temple stones have a box-shaped form. Exploitation of this feature allows a reliable segmentation of the stone models and enables a quick enumeration of all possible pairings. In the next step, using those pairings, a way of measuring similarity between two joints is introduced. To achieve this, several point-based methods are combined to allow for a reliable prediction. In the third step, the results of the previous step are converted into rank-based values and pair-wise matchings with a low rank get pruned to reduce the number of overall possibilities. Next, the remaining probable matchings are taken step by step to form larger solution parts, until no further suitable pairing is available. This fourth step is achieved by making use of tree structures. In the last step, by letting craftsmen and specialists include their profound knowledge on Khmer culture and temple sites, the solution proposed by the algorithm can be verified.

1.4 Contributions

A detailed analysis of existing works in the area of 3D puzzles and cultural heritage shows that none of the existing propositions is capable of piecing together Khmer temples based on their separate stones. In order to be able to solve such large-scale 3D puzzles, a classification of potential matches is needed. This calls for a method which can classify vertical and horizontal joining surfaces and is capabale to judge if two surfaces fit together. Furthermore, this approach must be able to distinguish between different vertical and horizontal joints quickly and assess which one is to use for a possible matching.

This is achieved by the first contribution: a new representation of the sandstone blocks developed in the course of this thesis, called *minimal volume box(MVB)*. It is a wireframe representation which resembles the outer shape of a virtual stone model using a minimum of storage capacity. The idea is that the sandstone blocks closely resemble the form of a cuboid (or polyhedron in a mathematical sense) making the MVB a polyhedron approximating the edges of the stone based on a linear-least-squares approach.

The second contribution is a similarity analysis especially tailored to Khmer temple stones. Before the actual puzzling can be started, it is of the utmost importance to gain reliable information on the probability of a vertical or horizontal joint. This is the same

as within the framework of classical 2D jigsaw puzzles. Only if two pieces are considered to fit they are tested for a possible match. Blocks of sandstone from ancient Khmer temple sites feature the special difficulty, that there are no mirroring counterparts which interlock. To resolve this drawback, a combination of different per-point based methods in 2D and 3D allows for reliable propositions on the probability of a match. Appropriate pairings will score highly in all measurements.

The third major contribution of this thesis is the five-step workflow to which the MVB computation is integrated in order to make use of the similarity analysis. This is the actual algorithm puzzling together the separate stones. Its first step is the MVB creation and determination of possible pairings, the second step is the already mentioned similarity analysis. After this, possible pairings are classified and pruned out if termed unprobable. The fourth step then combines remaining pairwise matchingsto complete solutions using a tree structure. In the last step the unique identification of the correct solution is determined by applying a force-directed graph drawing algorithm.

1.5 Outline of thesis

The thesis is structured as follows: Chapter 2 gives an overview of available 3D puzzling techniques that are the basis to solve the given task. Additionally, 2D puzzle techniques are reviewed as well, as it strenghtens the outcome of the puzzle algorithm to also take into account the 2D layout of each stone. Further areas of research that were investigated are presented as well. After this, chapter 3 talks about Khmer temples and temple building techniques and discusses how architectural insights can be exploited for a virtual reconstruction algorithm. Chapter 4 explains bounding volumes and derives a new type of representation for virtual temple stones, the *minimal-volume-box (MVB)*. Chapter 5 discusses possibilities how to compare two stones and develops a measure for the similarity of temple stones. The results are used for the reconstruction algorithm. Chapter 6 presents new methods to apply the similarity analysis for the reconstruction of the temple. The results of this new puzzle pipeline are discussed in chapter 7. A summary and an outlook on possible enhancements is provided by chapter 8.

Chapter 2

Related Work

> *Reconstruction of fractured ancient artifacts such as frescoes, pots, statues and tablets is important because it helps archaeologists make inferences about past civilizations and cultures. Unfortunately, reconstruction is usually a painstakingly labor-intensive job which may take several months or even years to complete by hand if the number of fragments is very large.*[6]

The aim of this dissertation is to develop an automatic virtual algorithm that is able to reassemble fallen Khmer temple walls. Those temple walls typically consist of individual stones and can be seen as a large virtual puzzle where each stone represents a puzzle piece. Due to the high complexity of this task it is beyond the scope of this thesis to abstract problem and solution to other temples as well. In general, puzzles can be divided into two or three dimensional puzzles. 2D puzzles are often called jigsaw puzzles, see figure 2.1 for an example. The reconstruction of the studied Banteay Chhmar temple wall can be tackled from the 2D as well as from the 3D perspective. In order to enhance the reliability of the matches proposed by the newly developed puzzle algorithm a combined 2D and 3D approach is used, although the focus is on 3D. This chapter will briefly talk about 2D puzzles and then elaborate on various methods to solve 3D puzzles.

[6]Shin, H., et al.; Analyzing fracture patterns in Theran wall paintings; **in:** Proceedings of the 11th International conference on Virtual Reality, Archaeology and Cultural Heritage; Eurographics Association, 2010; p. 71

2.1 2D puzzles

The earliest mentioning of modern-day jigsaw puzzles dates back to the 1760's when they were created as an educational aid for children to help them learn geography. In the industrial age jigsaw puzzle production became less expensive leading to an increased popularity and spreading. During the *"Golden Age"*[7] in the 1920's and 1930's the geometrical shape of each piece got more complex and thus the difficulty of 2D puzzles increased leading to an increase of adults solving those kinds of riddles as well. This increase in popularity caught the attention of scientists and in 1976 Garey and Johnson [GJS74] proved that 2D jigsaw puzzle belong to the class of NP-C problems, which is known to be very hard to solve.

Tybon [Tyb04] wrote an overview on history, problem statement and solution of jigsaw puzzles. He states the mathematical formulation of a 2D jigsaw puzzle as a rectangular grid with r rows and c columns, which form an array of quadruples, given in 2.1.

$$G = (r \times c \times 4), \tag{2.1}$$

where $G[i, j, k] \in \mathbb{R}^+$, for $i = 1, ..., r,\ j = 1, ..., c$, and $k = 1, 2, 3, 4$.
Each element of this array represents a value associated with one of the edges of a puzzle piece. Therefore, the four edges of a rectangular puzzle are denoted by $G[i, j, 1]$, $G[i, j, 2]$, $G[i, j, 3]$, $G[i, j, 4]$, where (i, j) is the position of the piece in the puzzle grid. Tybon proposes that the optimum solution (2.2) of a jigsaw puzzle is the minimization of the following sum:

$$\min \sum_{i=1}^{r} \sum_{j=1}^{c-1} |G[i, j, 3] - G[i, j + 1, 1]| + \sum_{i=1}^{r-1} \sum_{j=1}^{c} |G[i, j, 4] - G[i + 1, j, 2]|. \tag{2.2}$$

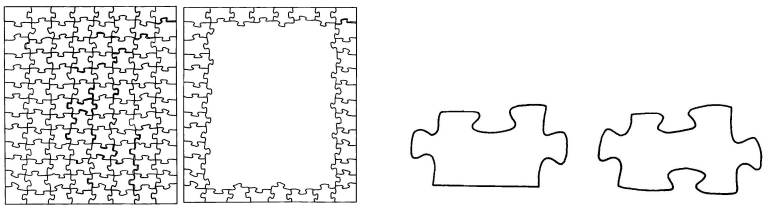


Figure 2.1: *A classical 2D jigsaw puzzle. A human would firt sort and puzzle the border thus reducing the problem size and then puzzle its interior. Source [WSKY88]*

[7] http://inventors.about.com/gi/dynamic/offsite.htm?site=http://www.jigsaw-puzzle.org/jigsaw-puzzle-history.html, accessed 01.08.2014

Generally speaking two types of 2D puzzles exist: pictorial and apictorial ones. The former makes (nearly) exclusive use of the information that is gained by examining the picture on each of the pieces and develops an algorithm which correctly combines the fragments. Apictorial puzzle approaches either cannot or do not want to use the pictorial information. In 1964 Freemand and Garder [FG64] gave the first and clear description of the characteristics (orientation, connectedness, exterior boundary, uniqueness, radiality) of apictorial jigsaw puzzles along with a suggested solution algorithm. Since then a lot of research in this area and the area of pictorial puzzles has been done. A survey concerning both types of 2D puzzles (pictorial and apictorial) is given by Kleber and Sablatnig [KS09], [KS10]. The authors also briefly comment on 3D puzzles yet they focus on the 2D case.

An additional puzzle research field are ripped up documents although their classification of being either a 2D or a 3D problem is not clear: In general it is agreed on that machine-ripped up documents belong to pictorial 2D puzzle types. The geometric form of such pieces is identic. Hand-ripped up documents are usually viewed as 3D puzzles. This is due to the fact that during the tearing possibly different layers of paper get broken. Unfortunately none of the features of ripped up documents occur on stone blocks therefore they will not be considered for further discussion.

2.2 3D Puzzles

Latest in 1971, over 40 years ago, when Hilaire G. DeGast got the patent for *three dimensional puzzles*, (see figure 2.2) the step from 2D to 3D was taken. In his patent application DeGast states that *"Three-dimensional puzzles are disclosed for forming a hollow ornamental object having a surface of revolution"* [DeG71].

In the academic context, scientific research started about 15 years ago with the works of Üçoluk [UT99]. To the best of the author's knowledge there are seven reviews done by Agapiou et al. [AGI+08], Eliuk and Boulanger [EB08], Kleber and Sablatnig [KS09], Kleber and Sablatnig [KS10], Willis and Cooper [WC08], Gomes et al. [GBS14], as well as Tsamoura et al. [TNP11] on 3D puzzles and cultural heritage in virtual reality.

In addition to these reviews the author would like to highlight the following works, giving an overview on different aspects of 3D reconstruction in the field of cultural heritage. From the abundance of works the ones featuring the highest similarity with the current work are selected and discussed. The literature section of Gomes et al. [GBS14] containing more than 100 published papers, provides a profound basis for further works in this large field of research for the interested reader.

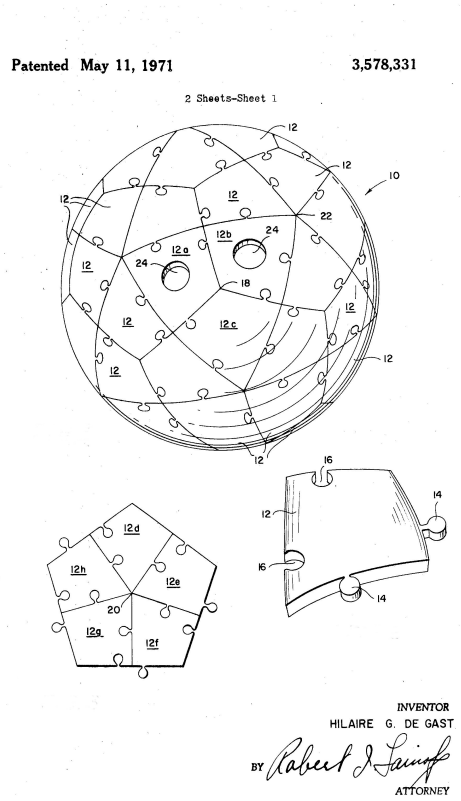
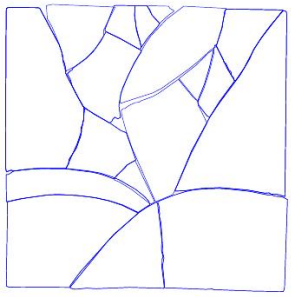


Figure 2.2: *A spherical 3D puzzle according to DeGast, [DeG71].*

One of the first works concerning virtual automatic reconstruction of three dimensional surface objects from archaeology was published in 1999 by Üçoluk and Toroslu [UT99]. They assume a hollow object, i.e. an object having no thickness. Therefore, it can be represented by a surface in Euclidean space. The boundary of such a surface can be viewed as a closed 3D space curve. Üçoluk and Toroslu propose an algorithm, using curvature and torsion of 3D space curves, that is able to automatically match fitting curves and thus reconstruct a broken object. Their algorithm is already capable of taking erosion into account which is an important feature.

In 2001 Kong and Kimia [KK01] approached the problem of 2D and 3D puzzle solving through curve matching as well. They propose a 2D costfunction measuring the similarity of 2D curves from puzzle pieces, see figure 2.3(a). The function is the sum of determining similarity in length of the curves under consideration and computing the difference in the rotation angle. The authors use a coarse-to-fine strategy, where all possible matches are first determined on a coarse scale and only get refined if the costfunction performs well enough. Thereby the algorithm is able to determine possible matches on different resolu-

tion scales and an impossible fit can be rejected at a very early stage. The costfunction and the coarse-to-fine approach then get generalized to 3D space curves, see figure 2.3(b). Both approaches rely on dynamic programming to be able to perform all computations in a reasonable amount of time. Their 2D approach is used in this thesis for the similarity analysis of two temple stones and will be discussed in more detail in chapter 5.



(a) *Automatically solving jigsaw puzzles using a 2D costfunction on a coarse-to-fine scale [KK01].*

(b) *Modifying the 2D costfunction to include torsion and extend it to the 3D case [KK01].*

Figure 2.3: *2D(a) and 3D(b) puzzle algorithm using (space) curves to virtually reassemble the object, both [KK01].*

Also in 2001 Papaioannu published *Virtual Archaeologist* [PKT01], a tool for semi-automatic reconstruction of threedimensionally acquired archaeological finds. It is divided into three main stages: 1) mesh segmentation (restrict search space to potentially interesting sides of fragments), 2) fragment matching (use matching error based on point-to-point distance derivatives) and 3) full reconstruction (find globally minimal matching error and match fragments).

Kampel and Sablatnig [KS03], [KS04] and Willis and Cooper [WC04] published works on how to virtually reassemble broken pottery. Basically, all make use of the fact that pottery vessels are symmetric to their rotation axis, yet they propose different methods of match finding. Kampel and Sablatnig extract profile lines, determine the rotational axis on each fragment and combine both information for a complete reconstruction. Willis and Cooper extract profile curves of each sherd fragment and use methods from probability theory to compute all reasonable alignments in form of a matching cost. The puzzling and merging of fragments is then done favoring those alignments resulting in a minimal sum of matching costs.

Probably the most noted work was published by Huang et al. in 2006 [HFG+06]. Using arbitrary fragments from non-symmetric broken objects they are able to automatically reassemble them as complete 3D models. Their reconstruction process consists of four

steps (see figure 2.4): 1) each fragments' surface gets segmented and classified into original (outer) surface and fracture surface, based on boundary curvatures and surface roughness, 2) a multiscale feature extraction based on integral invariants takes places, leading to 3) a pairwise matching of the fragments and finally 4) multi-piece matching using an iterative greedy algorithm.

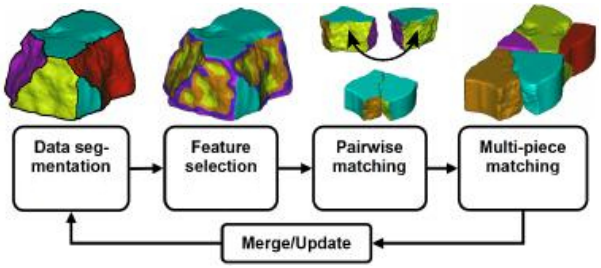


Figure 2.4: *The most prominent work concerning 3D puzzles has been done by Huang et al. [HFG+06]. Using a four step reconstruction process digitized broken objects are automatically reassembled.*

Winkelbach [Win06] and Winkelbach and Wahl [WW08] also work with non-symmetric objects and propose an algorithm which creates a series of probable pose hypotheses for each two fragments and employ a coarse-to-fine strategy using a cluster tree. This tree is then used to reject wrong matches at early stages, which saves computational time.

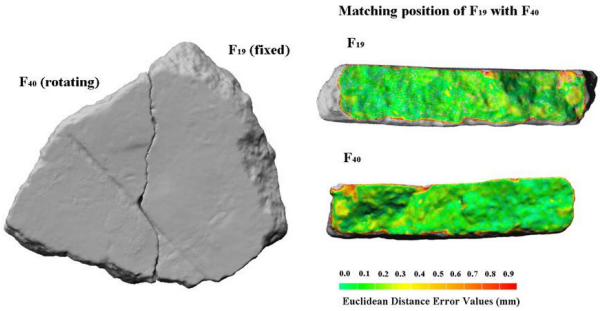
Parikh et al. [PSCC07], as well as Reuter et al. [RRS+07], Mellado et al. [MRS10] and Palmas et al. [PPCS] all work on semi-automatic ansätze to solve digital 3D puzzles. Parikh et al. [PSCC07] developed an algorithm that computes promising matching candidates between a query and candidate pieces in a database and displays them to a user. This algorithm is based on five sequential stages: 1) find a region that is of interest for the reassembly, 2) get local description of the interest region, 3) find near-neighbor local correspondences, 4) compute the geometric agreement and 5) analyze the match using a spectral technique based scoring. Reuter et al. [RRS+07] presented a user study for a semi-automated system which they developed. With this system a user is able to virtually reassemble broken pieces. Mellado et al. [MRS10] improved Reuter's system by adding a real-time geometric matching algorithm and a feedback to validate or refine the proposed match. The work of Palmas et al. [PPCS] is similar to the already described ones. The main difference is that in the last method an expert user defines several constraints based on his experience. Those are then taken into account for a reassembly where the algorithm finds the most suitable solution.

In a different approach Oxholm and Nishino [ON11], [ON13] attribute their 3D approach to 2D image registration. They extract the fragments' boundary as well as the

colour of each boundary contour. Then this information is encoded as a multi-channel 2D image represented as a graph. By determining cycles in this graph, the fragments are incrementally matched. The advantage of this approach is that no assumptions on geometry or structure of the object need to be made. On the downside this approach relies on user interaction in each step to validate if a match is accepted or not.

The series of papers published by B. Brown, T. Funkhouser, S. Rusinkiewicz, C. Toler-Franklin, H. Shin, et al. ([BTFN+08], [BLD+12], [FTFC+11], [SDF+10], [TFBW+10]) is an example for ongoing work concerning the virtual two- and threedimensional reassembly of fragmented objects (see figure 2.5(a)) in order to assist archaeologists. The objects are different broken wall paintings mostly from excavations in Greece, the Netherlands and Belgium as well as a synthetic fresco which has been professionally created and shattered such that a basic groundtruth is known. In Brown et al. [BTFN+08] the authors mainly focus on acquisition and processing of found fragments. They propose an acquisition workflow incorporating 2D as well as 3D scanning of data. After the acquisition, the 2D and 3D information is combined into one virtual model and a first matching based on ribbons is presented and discussed. Shin et al. [SDF+10] and Franklin et al. [TFBW+10] use the same data and extend the matching methods to gain further information. The former ([SDF+10]) elaborates on how fragment contours feature patterns that can be used for a matching algorithm. The latter takes into account a large variety of features, such as average color, contour curvature, average normal or dominant orientation. Brown [BLD+12] focuses on a visual approach. A software solution is presented that displays possible matches to a user, who is then able to accept or refuse the results. The authors of the last paper of this series, Funkhouser et al. [FTFC+11] take the same frescoes and apply a combination of the methods used in [SDF+10] and [TFBW+10] and apply them to a machine-learning approach. They suggest that, if the system has learned how to match one fresco, it can be used to automatically match other frescoes. The drawback is that these approaches present methods to help verify predicted (either by an expert or a computer) matches but unfortunately have not been applied to reassemble a fresco wall of unknown geometry.

(a) *Shattered pieces from a wall fresco. The picture shows how they are found at the excavation site [BLD+12].*



(b) *The broken pieces are scanned using 2D and 3D scanning methods and then reassembled on a computer [PAE+12].*

(c) *Final impression [PPE+02]*

Figure 2.5: *Virtually reassembling a broken wall fresco. (a) shows pieces found at an excavation site, (b) their digitized counterparts and (c) a virtually reassembled fresco.*

M. Fornasier [For09] also works on virtually reassembling broken frescoes. His dataset comes from the Ovetari Chapel in Padua, Italy. He proposes a 2D approach, based on circular harmonic decomposition to explain, where, how and why mathematics is needed in the restoration and reassembly of broken objects of archaeological interest.

C. Papaodysseus and his group are working on the reconstruction of fragmented wall paintings as well, see [PAE+12], [PPE+02]. For their earlier approach done in 2002 (see [PPE+02]) they use 2D digital photographs of a dataset from Greece and were able to reconstruct a part of this painting consisting of 936 pieces. From those images a contour line based on pixels is extracted and those lines are compared to each other. The smaller the overlap of the pixels and the smaller the region in between two contour lines which are in contact the higher the probability they match. Their more recent work, published in 2012 (see [PAE+12]), is based on 3D scans of an other wall painting from Greece. It applies the 2D approach to 3D meshes, i.e. they compare the volume between two contact surfaces, consider the overlapping area, relate the area in contact to the maximum allowed volume and check the length of the contact curves. Alas, for the 3D approach no completely reconstructed result is shown but it is mentioned that suggested pairings were confirmed by conservators.

Another example of 3D reassembly in cultural heritage are the works concerning the Severan Marble Plan also called *Forma Urbis Romae*[8], published by Koller, Levoy et al. [KL06], [Kol08]: *"The Severan Marble Plan was an immense marble map of Rome constructed in the early 3rd century during the reign of Septimius Severus and is a primary*

[8]http://formaurbis.stanford.edu/, accessed 30th of May, 2014

source of topographical knowledge of the ancient city."[9] The work group of professor Levoy at Stanford university started a digitizing campaign around 2002, where all accessible remaining pieces of this huge map were threedimensionally recorded and inserted into a database. Additionally the group proposed a technique for a virtual reassembly of those broken pieces, taking into account additional points such as structure of marble, incisions on the fragments and thickness of each piece. This way they were able not only to confirm already existing matches but also to find new ones, see figure 2.6 for an example.

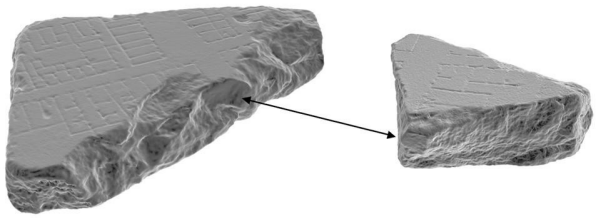


Figure 2.6: *Two fitting pieces of the Forma Urbis Romae project as published by Koller [Kol08].*

Approaches from G. Carra [CDV08], Geary and Howe [GH09], Oetelaar [Oet13], Sonnemann et al. [SSRS06], Arbace et al. [ASC+13], Ikeuchi et al. [IOT+07] and similar all use various methods such as photogrammetry, laser scanning, time-of-flight, to digitally acquire an object of cultural interest. These are for example the temples of Angkor Wat, the Lichfield Angel, the Baths of Caracalla or the Great Buddha. By using this threedimensional data in combination with a modelling or CAD software package[10], enhanced virtual models are created and missing pieces are modelled.

With regard to a threedimensional virtual reassembly of the Banteay Chhmar temple wall only the works of Winkelbach and Wahl provide helpful information as their cost-function is applicable to reassemble Khmer temple stones, see section 5.2.5. This is due to the fact that the preliminaries for the Cambodian temple site data set are different from the ones in the presented works. Temple stones are as such not part of a larger but broken object where clear incisions and heightenings provide highly probable fits. Additionally, the stones are not rotationally symmetric as they were quarried out and therefore do not provide symmetric features that could be exploited. Another hindering fact is that the proposed algorithms make use of special characteristics such as color or fragment thickness. This adds to the complexity of the topic as a completely new methods needs to be

[9]Koller, D. R.; "Virtual archaeology and computer-aided reconstruction of the severan marple plan."; Beyond Illustration: 2D and 3D Digital Technologies as Tools for Discovery in Archaeology, British Archaeological Reports International Series (2008): 125-134; p. 125

[10]e.g. AutoCAD, Blender

developed from scratch. To this end, the following sections will provide an introductional overview to the research areas from which the proposed puzzle solution workflow takes its algorithms.

2.3 Minimal bounding boxes

Khmer temple stones, which are the pieces of the present puzzle algorithm have a clear box-shaped form, see figure 2.8. A wireframe representation of their virtual counterparts enables a fast computation of matching possibilities and matching tests. To this end advances from areas such as reverse engineering, feature extraction and (tight fitting) volume boxes are discussed in this section.

The work of Mukherjee [MDC90] proposes an algorithm which is able to determine a wireframe structure of three-dimensional objects. The object under consideration is segmented into planar surfaces and thinned. Then each surface is polygonized and outliers are removed. With regard to simplifying Khmer temple stones this method was not applied as the prerequisites of this algorithms could not be met or modified to be met.

"Reverse engineering*, typically starts with measuring an existing object so that a surface or solid model can be deduced in order to exploit the advantages of CAD/CAM technologies.*"[11] An informative introduction to this topic is provided by Varady et al. [VMC97], Benko et al. [BMV01] and Werghi et al. [WFRA99]. Demarsin et al. [DVVR06] use reverse engineering to extract boundary curves represented as sharp features to segment the object.

Detecting (sharp) features (also known as *ridge valley lines*) in an object and using them to reduce the object has been investigated for long from different points of view. Boissonat [Boi84] makes use of nearest-neighbour-structures and Delaunay triangulation to extract a minimal representation of the shape of the object. Gumhold et al. [GWM01] convert the 3D point cloud into a weighted neighbourhood graph and create feature lines by extracting minimizing subgraphs. Ohtake et al. [OBS04] work in the same direction, yet they make use of first and second order curvature derivatives. Vosselman et al. [VGSR04] gave an overview on state-of-the-art methods of feature extraction in 2004, followed-up by Gross and Thoennessen [GT06] which propose an algorithm to extract lines from laser scanner point clouds. Fleishman et al., and Daniels II. et al. [FCOS05], [DIHOS07], [DIOHS08] propose to detect feature lines using robust moving least-squares approaches,

[11] T. Varady, R. R. Martin, and J. Cox, Reverse engineering of geometric models - an introduction, Computer-Aided Design (1997), p.255

smoothing the extracted lines and enhance them through spline representation.

Considering the construction of minimal bounding boxes or minimal bounding volumes the works of Barequet [Bar01], Bartz et al. [BKS06], Hu et al. [HFR10], O'Rourke [O'R85] and Suri et al. [SHH99] were important as a basis for the minimal volume box that has been developed for the temple stones.

All presented approaches are capable to reliably compute a regular box-shaped object consisting of planar surfaces which fit very closely to the underlying entity they are designed to resemble. They are able to simplify the virtual temple stones into box-shaped objects, yet a representation is needed that is not limited to planar geometry but that is also able to reflect the stone as closely as possible and segment the various side surfaces correct, fast and easy. Additionally, most of the methods are not able to detect whether an indentation is present or not and if yes, to locate it.

2.4 Mesh similarity

As already pointed out this thesis deals with solving a large scale 3D puzzle. Generally, in order to solve a puzzle on any scale or any dimension a criterion measuring the togetherness of two pieces is needed. In classical 2D jigsaw puzzles this criterion is usually expressed as correctness of the picture printed on each piece (pictorial puzzles). Yet, if the puzzling task is to fit together matching shapes (apictorial puzzle) a measure of similarity is needed making predictions on the geometric similarity of a possible match. In computer vision and graphics, the issue of *shape similarity* or *mesh similarity* has already been extensively studied, yet the meaning of *similarity* may differ in various works.

Commonly, given two computational representations of objects or shapes the research matter is to identify how much these entities resemble each other. To achieve this certain transformations such as translation, rotation or scaling can be applied to one of the shapes in order to match the other one as closely as possible partially or globally. This matching can be viewed as an application of *shape analysis* techniques. Shape analysis as such is the automatic examination of geometric structures in order to identify similarity e.g. of objects stored in a database. Important application areas apart from 3D puzzles include biology, forensics, CAD, or medicine. The Princeton Shape Retrieval and Analysis Group has intensively studied shape similarity, see [FK04]. *"Similarity assessment in 3D cases is usually carried out by generating shape signatures from the 3D models and then comparing these signatures using suitable distance functions. Ideally, these signatures should be representation independent and completely describe the features of the 3D model*

needed for similarity assessment. A shape signature could be a graph, a vector or an ordered collection of numeric values. The features captured by the signatures are usually dependent on the motivation for performing similarity analysis. "[12]. Important in all areas is an effective shape matching method allowing to evaluate the similarity. A survey on this topic is provided by Loncaric [Lon98] and Shum et al. [SHI96]. Further works in this research field include Adan and Adan [AA04], Alt [AG99], Barequet and Sharir [BS97], Chen and Bahnu [CB09], Kishon et al. [KHW91] and Milios and Petrakis [MP98]. Many of the works on different similarity features have various limitations such a restricting the shape of the matched objects, presuppose occlusion-free environments or limiting the motion [BS97].

In contrast to this *shape matching*, which usually occurs when working with a 3D acquisition device, can be considered as a form of similarity as well. Two 3D point clouds need to be identified and then registrated with each other. A good introduction to this topic is given in the dissertation of Simon Flöry [Flo10].

Techniques detecting shape similarity make use of a variety of methods such as geometric distances, topological features, geometric hashing, transformations in $\mathbb{R}^d$ space and are usually tailored specifically to the problem they aim to solve. Determining similarity in Cambodian temple stones lays the focus on a combination of different geometric distances, see chapter 5.

2.5 Packing problems

The sketch of the wall (fig. 3.12) which serves as a groundtruth in the current case (see chapter 3), the arrangement of the stones in combination with the task of puzzling them resembles a packing approach. In classical two dimensional *bin packing* a series of rectangles of different size is given and the task is to place them into a minimum number of bins of known size (see figure 2.7). The problem is known to be NP hard. Coffman et al. [CJGJ96] give a mathematical description of the one-dimensional bin packing problem as: *"We are given a positive integer bin capacity C and a set or list of items $L = (p_1, p_2, ..., p_n)$, each item p_i having an integer size $s(p_i)$ satisfying $0 \leq s(p_i) \leq C$. What is the smallest integer m such that there is a partition $L = B_1 \cap B_2 \cap ... \cap B_m$ satisfying $\sum_{p_i \in B_j} s(p_i) \leq C, 1 \leq j \leq m$? We usually think of each set B_i as being the contents of a bin of capacity C, and view ourselves as attempting to minimize the number of bins*

[12]Cardone, et.al.; A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications; J. Comput. Inf. Sci. Eng. 3(2); 2003, 110

needed for a packing of L. "[13] Jylänki [Jyl10], Dowsland and Dowsland [DD92], Berkey and Wang [BW87] and Dyckhoff [Dyc90] provide concise and informative overviews on various variants of the bin packing as well as strip packing approaches. Bin packing can be classified in either online or offline, where online means the number of rectangles is unkown in advance and the best configuration needs to be determined while the algorithm is running. Offline bin packing means the number of rectangles is known beforehand and the best configuration can be determined before the algorithm is applied.

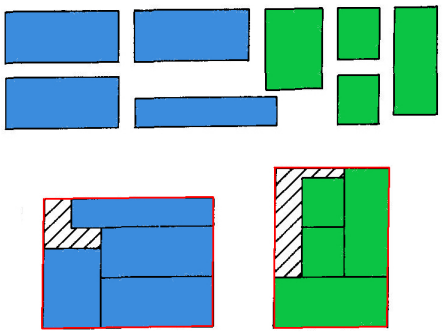


Figure 2.7: *Bin packing: The task of fitting different rectangles into bins thereby minimizing the number of bins used. In this figure eight rectangles were fitted into two bins. The bins have been redrawn in red to enhance visibility. Picture source: [Dyc90]*

Variants of bin packing are *strip packing*, *free-form strip packing* or *cutting stock*. In all three cases, there is only one bin of known width but unkown height (the strip). The objects here are not necessarily rectangles but are allowed to take any form needed. Such problems can occur e.g. in textile industry where clothes are cut out of the fiber.

Concerning a partial reassembly of the Banteay Chhmar temple that is dealt with in this thesis, the stones could be viewed as the rectangles and the wall as the bin the stones need to be placed in. As some of the stones have indentations, thus being non-convex polytopes and not rectangles prevents the application of bin packing methods as those are especially tailored to rectangles. Additionally, the width and height of the wall is unknown as it is indecisible in advance if the stones come from one vertical row or one horizontal column of the wall, from a rectangular part or neither. Thus it could happen that the algorithm would falsely indicate a one row or one column solution as correct.

An alternative to bin packing is polyomino packing. An n-omino is a 2D object made up from n squares. The most popular n-omino is the domino. Conway [CL90], Marshal [Mar97], Golomb [Gol70] and the book *Polygons, Polyominoes and Polycubes* [Gut09] give overviews and introductions to that area of research. Concerning the large scale 3D

[13] Coffman, Garey, Johnson; Approximation algorithms for bin-packing – an updated survey; p.46

puzzle of Khmer sandstone blocks polyomino packing or rather polyomino puzzling does not provide helpful insights as it is not possible to tile the irregular sandstone blocks into regular n-ominoes.

Another possible approach which is related to bin packing is *very-large-scale integration (VLSI)*. It places several thousands of transistors optimally on a single chip and creates an integrated circuit. VLSI design is a methodology for saving microchip area using optimization methods to minimize the interconnected fabrics area. Placement in this case is the process of determining the optimal location of the devices. Main objectives in the design flow are 1) to minimize area and depth, 2) to minimize wirelength, 3) to minimize delay and power, 4) to minimize power subject to timing or 5) to minimize changes. Unfortunately, none of the prerequisites for placement in VLSI design could be met or adapted to match Khmer temples stones. Yet, for the gentle reader, Chu [Chu08] gives an informative introduction on this topic.

2.6 Virtual reassembly by hand

In order to develop an automatic virtual reassembly algorithm a virtual manual approach was conducted to see where difficulties might occur. Figure 2.8 shows 61 stones that have been assembled manually using the Breuckmann software Optcocat. The software comes along with the 3D scanner which was used to digitally acquire the separate stones, see chapter 3. Throughout the manual assembly the sketch of the wall in its original position was used as a groundtruth. It took the author one week (about 40 hours) to complete this model, which means the reassembly needs to be tackled with sophisticated algorithmic approaches. The manually reassembled model has a size of about 2Gib and it is rather difficult to insert further stones. In addition to that the red circles show parts, where accumulation errors occured. This is due to the fact that by puzzling the stones from top to bottom using the sample solution all the small errors made by not perfectly aligning the stones sum up and in the end lead to large holes. Meshlab was used to visually enhance the visible parts. Before investigating and presenting methods for an automatic reassembly sound knowledge on how the Khmer build their temples is needed. This is presented in the following chapter.

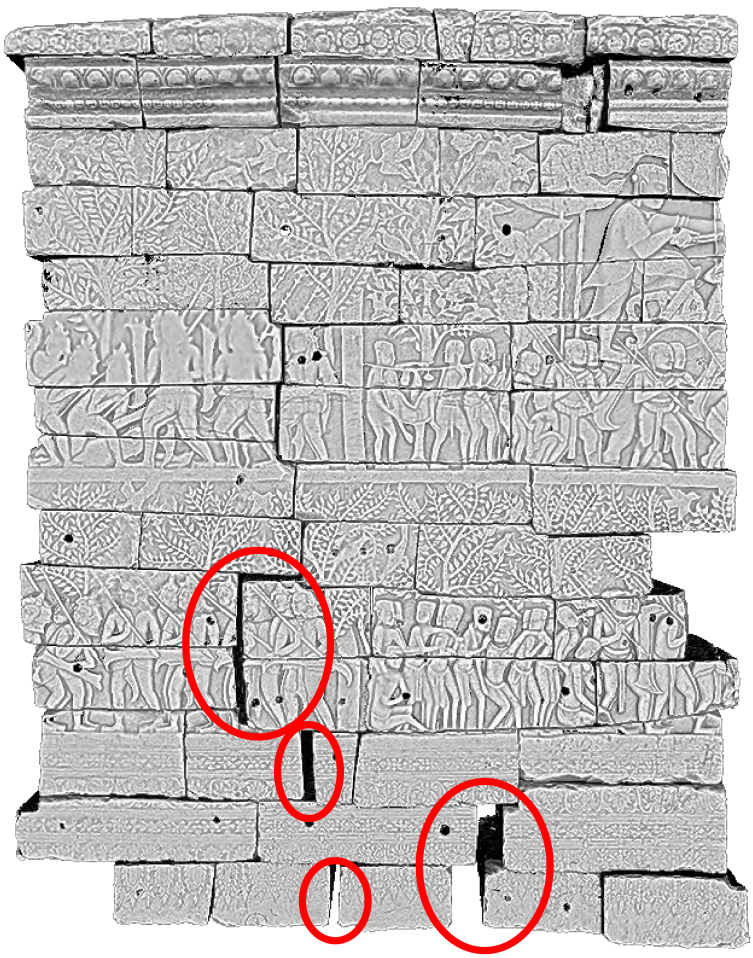


Figure 2.8: *A solely manual approach to virtually reassemble the temple wall. This model was created using the optocat software package that is distributed with the 3D scanner that has been used to acquire the separate stones.*

Chapter 3

Khmer Temples

> *Today, natural decay and the ravages of vandals and thieves (...) have left their mark, but no one who visits Angkor can fail to be stirred by its grandeur.*[14]

With regard to a virtual reconstruction and before an algorithm can be developed it is important to gain profound knowledge and a deeper understanding on how Khmer masons built their temples. This chapter gives a short overview on Khmer temples and will then focus on how they were constructed. From the fact that Banteay Chhmar and Bayon, a temple at the Angkor archaeological park, were both built by king Jayavarman VII, information on Khmer temple building can be learned. The Bayon temple features bas-reliefs with many scenes from daily life and amongst them also workers carving bricks.

According to Freeman and Jacques [FC06] Khmer temples are not, as their European counterparts (= churches) congregation places for faithful believers but rather an exceptional palace of a god. This leads to the fact that there was no need for vaulting or creating large interior spaces. The earliest temples were built around the sixth or seventh century A.D. With respect to their artistic criteria and construction period art historians can classify Khmer temples into 15 styles, see Nguonphan [Ngu09]. The temples usually have a quadrangular shape and are built by following various geometric rules. Another important feature is the moat surrounding a temple side. For more information on Khmer temple styles the interested reader is referred to the PhD thesis of Pheakdey Nguonphan [Ngu09] or the papers of Freeman and Jacques [FC06]. The following chapter

[14]Tully, J.; A short history of Cambodia; Singapore, 2005; p.30

is based on the works of E. Aymonier [Aym04], A. le Bonheur [Bon95], O. Cunin [CB05], J. Dumarcay [Dum73], [DR01], M. Freeman [FC06], C.F. Higham [Hig01], [Hig04], E. Lajonquiere [Laj11], D. Rooney [Roo06] as well as J. Tully [Tul05].

3.1 Banteay Chhmar

> *"Foreigners don't go to Banteay Chhmar, for good reasons. The ancient Khmer temple sits on the lawless northern frontier of Cambodia, throttled by malaria-infested jungle. The area is land-mined, banditry is rife, and the roads are barely passable. In short, Banteay Chhmar is a tempting destination only for those who like to get way off the beaten path."*[15]

(a) *The map of Cambodia shows its location within the country close to the border of Thailand as well as the famous Angkor Wat.*

(b) *The picture documents the state of deterioration inside the inner walls.*

Figure 3.1: *One of the largest known Khmer temples built by king Jayavarman VII is situated in Banteay Chhmar, Cambodia.*

Originally built under the reign of king Jayavarman VII ruling from his ascension to the throne in 1181 until his death around 1220 AD to honor his son and four of his army generals for defeating the Cham, Banteay Chhmar is nowadays gradually being reconstructed, see Higham [Hig01]. Due to its remote and isolated location, it had been nearly forgotten and the remarkable bas-reliefs had been subject to severe looting. Today, the temple region is slowly developing into a tourist attraction, despite the fact that Banteay Chhmar lies within a large minebelt stretching over several hundred kilometres along the Thai-Cambodian border. The map in figure 3.1 (a) shows the location of Banteay

[15] D. Preston; *The temples of Angkor. Still under attack*; National Geographic, 2000, vol. 198, no. 2; p. 86

Chhmar within Cambodia in reference to the famous and well-known Angkor Wat.[16] Despite the fact that the Khmer empire stretched across nearly all of former Siam at its height and that its trading networks even reached up into China, their only legacy are such astounding temple sites, spreading over all of the region.

The site

The templesite of Banteay Chhmar is located about 63km north of Sisophon and 20km east of the Thai border, close to the Dangrek mountains, and belongs to the Thma Puok District (see figure 3.1 (a)). As the romanization of Khmer writing can be ambiguous, its name means either *citadel of the cats* or rather *Narrow Fortress*[17,18]. The inner complex is enclosed by walls, which span an area of 2.2×2.4 kilometres, and is encircled by a moat. This moat has been fed by a stream from the nearby baray[19] in the northeast and is crossed by four fortified bridges. Those passages are bordered by sitting giant statues, similar to the ones at Angkor Thom as O. Cunin remarks, see [CB05]. Additionally, there are eight small satellite temples surrounding the moat, thus all in all Banteay Chhmar encompasses an area of about nine square kilometers, see fig. 3.3, making it one of the largest temples in the world. C. Higham states that *"This extensive area between the outer wall and the moat and walls of the inner temple, which covers 448 hectares, now includes only eight single-chambered shrines, but presumably at one time it would have housed a considerable population."*[20]. Some of the small temples are completely collapsed by today.

Inside the inner sanctum the walls are covered with bas-reliefs showing battle scenes between the Khmer and the Chams (see fig. 3.5) and scenes of daily life, very similar to the ones that can be admired at the Bayon temple, which is located close to Angkor Wat. Additionally, a lot of apsaras[21] and *"an extraordinary range of gods with multiple heads and arms"* [22] can be seen. The inner complex is a labyrinth of passageways and shrines.

[16]Map source: "Banteay Chhmar," Wikipedia, The Free Encyclopedia, http://de.wikipedia.org/wiki/Banteay_Chhmar (accessed May 12, 2014).
[17]Aymonier, E.; Le Cambodge. Les provinces siamoises, Vol.2, Paris, 1904, p. 335
[18]Inventaire descriptif des monuments du Cambodge, Vol. 3; PEFEO 9, E. Leroux, Paris 1911, p. 391
[19]A baray is some kind of water reservoir.
[20]C. Higham. The Civilization of Angkor. University of California Press, 2001, p. 131
[21]In buddhism and hinuism an apsara is a spirit of clouds and waters.
[22]C. Higham. The Civilization of Angkor. University of California Press, 2001, p. 131

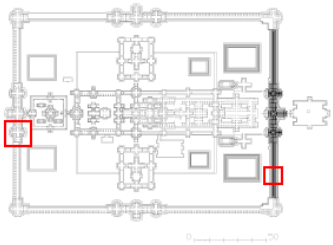


Figure 3.2: *A map of the inner temple complex. The red box on the right marks the part this thesis deals with, the red box on the left marks the Avalokeshvara (see fig. 3.4)*

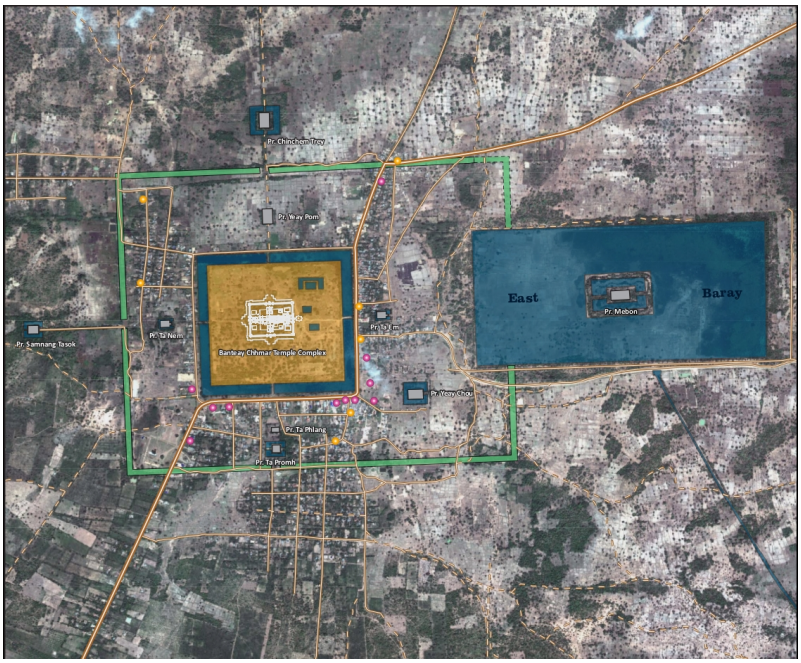


Figure 3.3: *Map showing outline of Banteay Chhmar, Cambodia. Picture courtesy of Global Heritage Fund: www.globalheritagefund.org*

The sanctuary's remote location and the peculiar building technique that does not use mortar or other alternatives caused about 80% of the templesite to collapse due to tree and foliage growing on the walls and rainwater ingressing every fracture, causing stones to shift. Of the remaining 20%, most structures require stabilization that keeps them from collapsing. Unfortunately, there is not much written information left of the Khmer, since their writing was done on palm leaves or animal skin, which decomposes quickly, see Tully [Tul05].

The site is similar to the better-known Bayon temple, which is situated in the Angkor complex, especially concerning the numerous towers showing faces of Jayavarman VII both sites feature. The reason for this is that Banteay Chhmar as well as Bayon were founded by the same king, Jayavarman VII. After his reign, Banteay Chhmar has been

given up and is therefore today considered as an unmodified historical monument. Thus, as Dr. Olivier Cunin remarks,[23] a better understanding of Banteay Chhmar and Bayon can be reached by studying both structures.

(a) *The combination of beautiful carvings and remote location encourages looting: This is a picture showing eight multi-armed Avalokeshvaras. The picture was originally taken by George Groslier in the 1930s when he was examining Banteay Chhmar.*

(b) *Status quo of (a) in 2010 from the back side. After severe looting in the 1990s, only two Avalokeshvaras remain. Two can be found in the National Museum in Phnom Penh, two are supposed to be found in the pile of stones in-situ and the remaining two are still missing.*

(c) *Front view of the last the two Avalokeshvaras that are still upright.*

Figure 3.4: *Beautiful carvings at Banteay Chhmar.*

The aforementioned remote location and the beauty of the carvings caused looting and severe damage to the Banteay Chhmar site. The most prominent example for this is a bas-relief, for which the temple is known best, the Avalokeshvara, a multi-armed buddhistic boddhisattva, also known by names such as Lokeshvara, Avalokiteshvara or Lokiteshvara. In 1998 parts of this wall were removed *"by the Cambodian army for sale on the Bangkok antiquities market."*[24].

[23] "The Small Citadel": Reconstructing the Ruined Buddhist Complex of Banteay Chhmar. Aspects of Angkor Lecture Series, Smithsonian Institution; 2010

[24] "Banteay Chhmar", in: C.F. Higham. Encyclopedia of Ancient Asian Civilazations. Facts on File Inc., 2004, p. 39

Figure 3.4 shows the remainingss of this bas-relief today. Of formerly eight remarkable Avalokeshvaras only two are still standing today. Figure 3.4(a) shows how they have been found by George Groslier in the 1930s, whereas figure 3.4(b) shows the status quo of the same part of the temple from its back side in 2010. Figure 3.4(c) is the front view of the last two still standing and most prominent Avalokeshvaras of Banteay Chhmar.

Figure 3.5: *One of the many battle scenes at the Banteay Chhmar temple complex.*

A computer-modelled reconstruction (see figure 3.6) of the templesite was done by Olivier Cunin, who studied the carvings and face towers of the Banteay Chhmar temple for over a decade. For more information the author refers to Sharrock, Jacques and Cunin [SJC], as well as Cunin [CB05].

Since 2008 the Global Heritage Fund (GHF) is leading a preservation campaign, which involves, among others, teaching locals how to conserve the precious bas-reliefs and stones. Supported by the Cambodian Ministry of Culture and Fine Arts, the IWR (Interdisciplinary Center for Scientific Computing of Heidelberg University) and the GHF are collaborating in order to develop appropriate techniques to support the conservation process using advanced digital acquisition and reconstruction techniques.

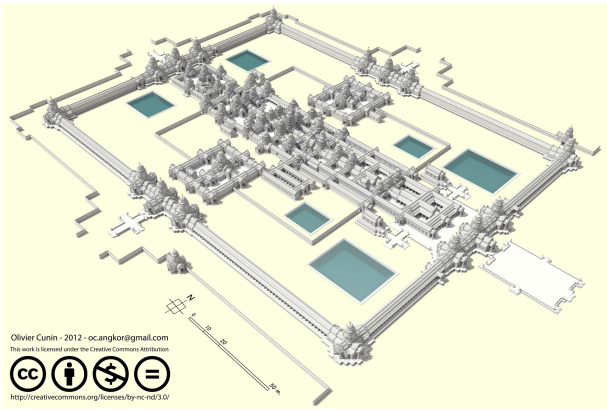


Figure 3.6: *A computer-modelled reconstruction of the Banteay Chhmar templesite done by Olivier Cunin. Picture courtesy of Olivier Cunin.*

3.2 Temple building techniques

Most of the currently available information on constructing a Cambodian temple site was gathered by the early French explorers travelling through then Indochina and from the analysis of the bas-reliefs on the walls of different temples. In figure 3.7. for example workers are depicted carving bricks.

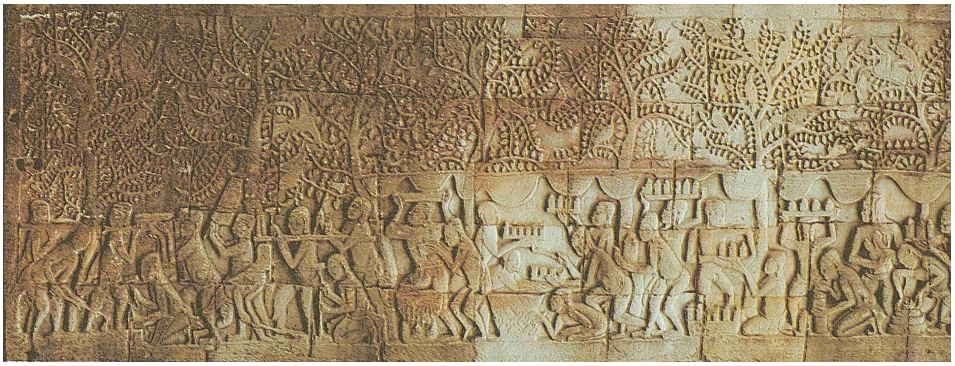


Figure 3.7: *Part of the southern wing of Bayon temple showing workers carving stones for the construction of a temple. Picture Source: [Bon95]*

In order to construct a monument like a Khmer temple, building material and an economic surplus is needed. Materials used in ancient south-east Asian complexes were mainly brick, sandstone, and laterite and, to a lesser extend, wood which by now is nearly completely deteriorated in most places. As the use of sandstone requires higher masonry skills and more manpower the earliest temples dating back to the eighth and ninth century were completely made from wood, followed by brick. Temples being in brick mostly have stucco facades, although sometimes it was carved directly. Stone however made a gradual appearance and for the Khmer case can be dated to the late 10th century. For every temple the considerable distance from the quarry to the construction site also needed to be covered – this is especially the case for sandstone, whereas laterite quarries were mostly exploited close to the construction site.

Quarries and Transportation

The quarries for most Khmer temples were located at the Kulen plateau, about 30km away from the central Angkor complex. For the temple complex of Banteay Chhmar it is not sufficiently known where the stones came were sourced yet in all cases the following principle was applied.

Within a quarry the sandstone was cut and shaped into blocks of various sizes, al-

though in a mathematical sense they are not blocks as they do not feature angles of 90°.[25] This cutting and shaping was reached by marking out a shallow vertical working face of roughly two metres, sometimes less. Next the block was shifted vertically and raised horizontally. While the exploitation of the quarry moved on the working face became higher and higher - yet the highest appears to have been five metres, probably because above that it was too difficult to shift blocks on rollers. The disadvantage of this technique is that it does not allow the blocks to be used vertically other than as a false bedding.

Another technique which has been used, hollows out a recess line and forces wooden wedges into it. Those wedges were then soaked in water for several days until the block split off along these markings.

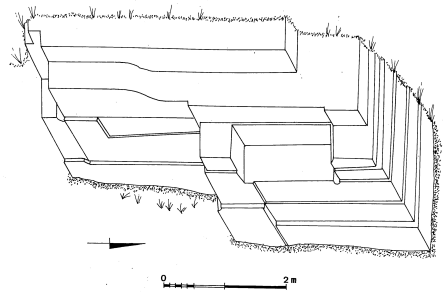


Figure 3.8: *The figure shows the cutting face of a sandstone quarry at the Kulen plateau. Picture Source: [DR01]*

To move the material from the quarry to the site the blocks were transported by elephant or ox carts, depending on the size. To hoist the stones, pairs of bamboo pegs were driven into specially prepared holes (two sets per block, see figure 3.9 for an example) and linked by ropes using tripods, leavers and pulleys.

Figure 3.9: *On the right side of the stone two holes are visible which were needed in order to be able to hoist the block into place.*

[25]It is known that the stones were quarried according to demand as the inventory of stones needed for building was drawn up on some of the temples.

Construction

After the material had reached the construction site the building of the themple could start. In general, the basic layout for construction was a base or a platform. If the temple was intended to represent a mountain, the masons formed a high platform using laterite to shape it and then filled it with rammed earth. In most cases sandstone was used to cover the laterite. In a temporary shelter close to the temple site the preparation of the blocks was undertaken by cutting them to the desired dimensions and grounding the layers against each other. This is nicely shown by a relief on the internal gallery at the Bayon, see figure 3.10. The bas-reliefs were sketched by the master carvers making the design. After everything was in place hundreds of carvers finished the sketch, which is indicated by a lot of unfinished examples and the final reliefs bear no relationship to the junctions of the stones they are carved on to.

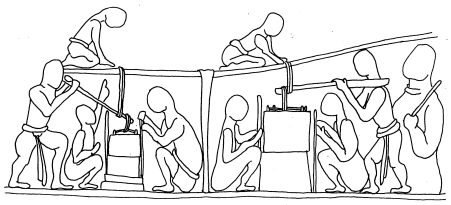


Figure 3.10: *Sketch of a bas-relief located at the internal gallery of the Bayon temple showing workers cutting stones. Picture Source: [DR01]*

During the construction process the stones were very seldom keyed together but vertical joints were laid out, one on top of the other. This created a weakness leaving tree roots space for infilration which then tore the structure apart. Additionally, the Khmer masons did not use any mortar or binder to fix the stones, just occasionally some metal clamps, first made of bronze and later iron. As metal is a very valuable commodity, most of them are stolen nowadays. In conclusion this means the only stability in the structure came from the cut of the stones. New stones were added in rows from bottom to top and if a stone was to be placed but did not perfectly fit, then an indentation or kink from the underlying stone was cut out.

Rooney [Roo06] states that after the stones were set in place, they were ground together using abrasive sand between them and rocked back and forth until a tight joint was formed. This is not accepted by all scientists and there is another theory how the stones were prepared. Dumarçay e.g. [DR01] doubts Rooneys opinion, stating, the stones were first roughly cut out, finished with a chisel and then polished. His argument is that, assuming stone A is grinded against stone B on the same height as the moving stone

(A) where the vertical and the horizontal surface can be processed, if stone A is smaller than stone B, the surface of B being grinded by A is larger than the grinding surface over the whole length. But nothing similar can be seen at the Bayon, only the contact surfaces have been worked on. Furthermore Dumarçay suggests, that a wall is separated into several parts stabilizing each other. He is convinced that the assembly was as follows: the first layer was set in place and blocked at an edge. The cornering stones were placed higher and adapted to form an offset in which the second layer was fit into, which in turn was also blocked by an edge. The free space at the corner was filled with specially prepared stones before the third layer came, and so on. This scheme has not always been pursued, as can be seen by missing edges or missing offsets. Additionally, the described scattering technique explains the sole working on nearly horizontal surfaces.

An explanation not mentioned by Dumarçay but facilitating his arguments is that that the grinding of the stones is not likely to have been taken place in situ since the depth of the walls is not always constructed from a single block of stone. This way when the stones were placed they needed to by adjusted according to their cut of stereotomy which is sometimes very complex.

From the ninth century on Khmer masons started to employ stone wedges in the vertical plane and later also in the horizontal plane to be sure of the coherence of the masonry courses, see figure 3.11.

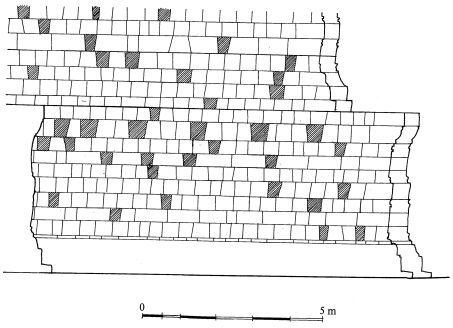


Figure 3.11: *One of the walls from the Bayon temple where the wedge stones, wich are used ensure the coherence and stability of the wall have been marked in black. Picture source [DR01]*

According to Dumarçay the wedge repells the stress beyond the length of the lintel and avoids straining this element too much as it only rests on top of the jambs on each side. Thus the wall gained a greater solidity and undertook the structures with a greater verticality showing that the Khmer adapted to the sandstone material in the course of time. The wedges can also be found on the wall segment that serves as a groundtruth for this thesis, see figure 3.12.

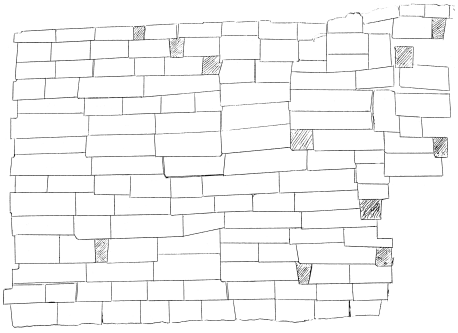


Figure 3.12: *Sketch of the wall in the Banteay Chhmar temple, that has been threedimensionally acquired.*

3.3 Data

The 3D puzzling task of a fallen Khmer temple side this thesis deals with is based on separate stones from the Banteay Chhmar temple. Therefore, those stones were acquired using digital measuring equipment. In the following, a short overview on 3D acquisition techniques is given and the data acquisition in-situ summarized. For further information on machine vision the reader should refer to the Handbook of Computer Vision and Applications, volume 1 [JHG99].

In recent years digital data acquisition has gained increasing interest in the academic context. The equipment got more affordable and new insights in existing research as well as new research areas are thus possible. Topics of investigation are for example the indentification of wedges on cuneiform tablets[26]. One of the goals there is to enhance the readibility of the writing. Another example is under-water archaeology: using photogrammetric methods, i.e. 3D information is gained by using 2D photographs taken from different viewing angles, objects on the sea bottom can now be studied without necessarily hoisting them, see Drap et al. [DSS+07], and Drap [Dra12]. Further applications are models gained by aerial photographs, see Altan et al. [ACKT04], digitally acquired cave paintings, see Stanco et al. [SBG11] or adding models into databases, see Koller and Levoy [KL06]. Mixed methods open up additional possibilities: using threedimensional scans of ruins in combination with virtually modelled objects, whole cities can now be explored on a computer, see Dylla et al. [DFM+08] or simulations answer questions as to the reason of a specific placement of a building, see Frischer and Fillwalk [FF13].

Threedimensional digital data acquisition equipment can acquire two different types

[26]For further information see to the dissertation of Mara, [Mar12]

of data: i) hollow surface meshes where only the outer hull of the object is acquired and ii) volumetric depth image where focus is laid on acquiring the inside of an object. Data from the latter type is mainly dealt with in medicine (e.g. tomographic data) and will be skipped from further consideration in this thesis. The former type of data is used throughout this thesis and will be considered in more detail in the following section. Digital surface models can be acquired using

- Triangulation:
  The object under consideration is viewed from two different known viewing points. Depth information can be gained by the difference of both images.

- Running time:
  A signal with known velocity is emitted and reflected by the object. Depth information is gained by the time it takes the signal to travel to the object and back.

- Interferometry:
  Depth information is gained by measuring differences in amplitude and phase of an emitted ray.

- Shading or structured-light:
  A pattern of black and white stripes is projected onto the object and at every transition from black to white the scanner gains depth information about the object.

- Photogrammetric stereo:
  Using a set of twodimensional images from a camera taken from different (unknown) viewing positions depth information is gained by the difference in the photographs.

Additional techniques are usually a variation or combination of the just listed types.

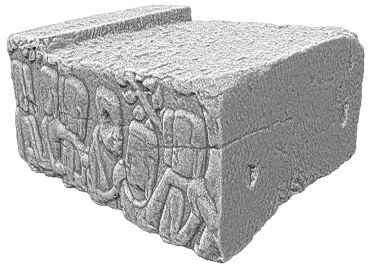
The different scanning methods feature several advantages as well as disadvantages, which can be separated into pricing, range and accuracy. Accuracy in case of digital measuring equipment means the potential of representing the object under consideration as perfect as possible reducing the error on the surface to a minimum. Range is related to the size of the object which can be measured. E.g. a time-of-flight scanner (ToF; running time type) can capture objects up to 300m in size. Not every method is suitable for every application. E.g. photogrammetry is usually extremely cheap as it uses digital images, which can be acquired with an ordinary digital camera at any place and at any time and any object and there exist free software packages for depth calculation. Its disadvantage yet is the lack of accuracy especially for very small objects. On the contrary shape from shading and triangulation methods have the possibility to acquire objects with

an accuracy in the μm scale yet they are very expensive as highly sophisticated hardware and software equipment is required.

The technique used for the data acquisition this thesis is based upon is a mixture of structured-light and sterevision. To gain a very high accuracy the structured light pattern size is varied at the power of two and additionally a phase shift is done. The combination of stereo vision and structured-light techniques leads to a precision of up to 10 μm. Furthermore, the scanning device can use its two color cameras to acquire colored texture maps, thus gaining a more realistic impression of the object. Figure 3.13(b) shows one of the stones during data acquisition, with the typical stripe pattern, caused by the acquisition process.

(a) *A stone after the cleaning process and before being digitally acquired.*

(b) *A stone with the typical stripe pattern of the 3D scanner.*



(c) *After some postprocessing steps, a complete model of the stone can be viewed on the computer.*

Figure 3.13: *The 3D data acquisition process in brief.*

In 2010 a team of the *Interdisciplinary Center for Scientific Computing (IWR)* digitally acquired different parts of this temple with the aim to provide groundtruth work in the area of large-scale 3D Khmer temple puzzles. The data that is used in this thesis as a groundthruth is based on 135 digitally acquired stones from a part of Banteay Chhmars' eastern outer temple wall from the inner complex, cf. figure 3.1(c). Figure 3.14 shows this wall in its original state in December 2009. The acquired stones are the ones between the two red lines. The foundations of the wall in question needed to be reinforced to enhance its stability and prevent it from deterioration. Therefore, a sketch (see figure 3.12) and photographs of the wall were taken, and all stones were labeled while being taken down. Due to several cirumstances it was not possible to scan during nighttime in Banteay Chhmar, yet structured-light scanning techniques need a very dark environment to operate at their best. To overcome this a dark tent was used to perform the data acquisition during daylight. Further data acquisition details, applications and outlook of the projects can be read in the papers by Schaefer et al., [SMF+11] or Freudenreich et al., [FSN+11].

Each of the digitally-acquired stones as well as the fallen blocks still lying around

consists of sandstone with a unique geometrical structure. On site, the Global Heritage Fund already began a manual reassembly, but taking into account the weight of each stone (up to one ton) as well as the fragility of the material, the abundance of fallen stones and the climate conditions, a computer-aided documentation and reconstruction process was favoured.

The fact that the aforementioned eastern wall had to be taken down stone by stone, proved a great possibility to capture individual stones and investigate *"whether a virtual reconstruction based on scans is possible in order to fasten the reassembling process and enhance the safety for manpower and stones"* [27].

Based on the digitally acquired data, characteristic features of each stone can be analyzed in order to define criteria necessary to verify if the blocks are adjacent. Automating this virtual 3D puzzle is of great help for archaeologists and stone conservators reconstructing the temple site.

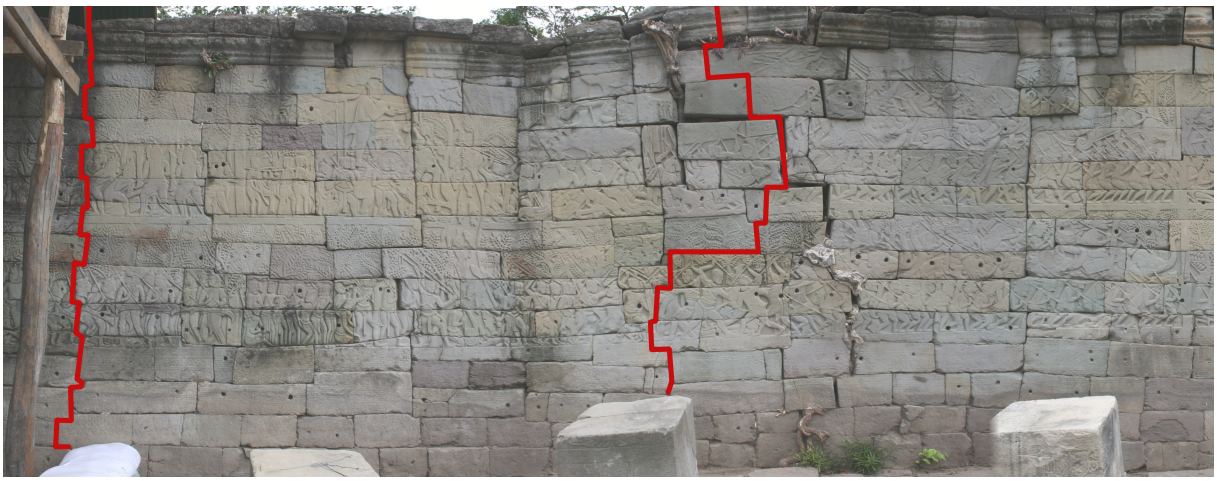


Figure 3.14: *The set of stones that is used throughout this thesis.*

3.4 Optimization and Khmer temple reassembly

The task of reassembling the bas-relief wall from the Banteay Chhmar temple site requires the assignment of digitally acquired stones to the position they had before the wall was dismantled. This can be described as a problem from the area of optimization: in order to find the original (= correct) positioning of each stone either a similarity analysis can be conducted to determine the positions via algorithmic methods or an exhaustive trial and error approach can be applied via positioning each stone until the correct solution is found.

[27] Freudenreich, J. et al.; Close range architecture documentation of angkor style temples; in: Proceedings of DMACH 2011; p. 239

Due to expert knowledge from stone conservators and other Khmer architecture experts, it is assured that the correct solution is unique. Independent from the puzzling method of choice (complete enumeration, algorithmic methods or a combination) the problem is to find the optimal solution from many possible. The stones have been worked on such that they fit very tightly, i.e. the objective function of this optimization problem needs to be able to detect pairings of stones featuring the highest similarity and determine the optimal layout.

The problem can be classified as being a combinatorial optimization problem with a constraint. Given a pair (S, f), where S is a finite set of all possible pairings of two stones and f is a function assigning similarity values to pairings of stones, the goal is to find a globally optimal solution $i \in S$, such that s minimizes f, i.e. $f(s) = \min_{x \in S} f(x)$. The constraint is that the stones may not intersect, i.e. the Euclidean distance $d(k, l)$ between two stones k, l is always required to be greater than zero.

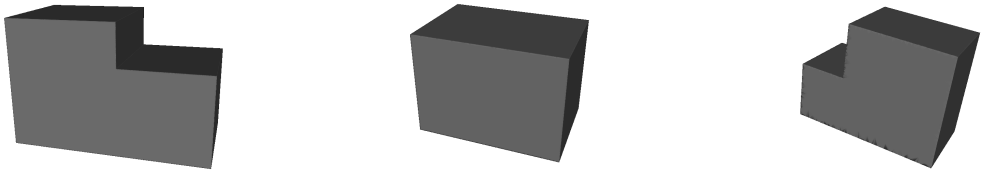
Let $1 \leq i, j \leq n$ be two stones and t_i, t_j their respective sides. A pairing $p = (i, t_i, j, t_j)$ thus constitutes a potential match of two stones describing how the position of i and j relates to each other. A similarity analysis (given as 3.1) assigns each pairing p of such two stones a similarity value r_s.

$$\tau(p) = r_s \tag{3.1}$$

A solution for assembling n stones therefore consists of at least $n - 1$ pairings. Thus, the optimal solution can be given as

$$\min \sum_{k=0}^{n-1} r_{s_k} \tag{3.2}$$

As a proof of concept, this has been tested for a synthetic dataset, see figure 3.15. Three stones with typical features of Khmer temple stones as described in this chapter were designed and the puzzle workflow applied. In this ideal case the optimal solution of (3.2) is the correct solution, see figure 3.16.



(a) *Stone with an indentation to the right*

(b) *Stone with no indentation*

(c) *Stone with an indentation to the left*

Figure 3.15: *A synthetic dataset of Khmer temple stones.*

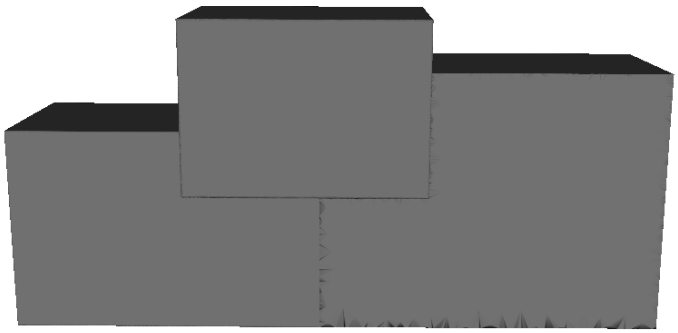


Figure 3.16: *The optimal solution of the synthetic dataset is the correct solution.*

3.5 Conclusions for a reconstruction algorithm

Concluding chapters 1 - 3 the given task is to virtually re-erected a fallen Cambodian temple from the Angkorian period by solving problem (3.2) and the development of an as automatic as possible algorithm. Chapter 1 outlined the need of a computational method to help people on site: the assembly can be done much faster, the fragile stones are carried around less and the security of the workers is enhanced. Chapter 2 showed that a completely new and indepented approach needs to be developed, as already existing work is not useful in the current case. It additionally pointed out, that the amount of data increases fast and small errors accumulate. On top of that, a purely manual puzzling done with the computer provides no gain in time compared to the manual puzzling on site. Chapter 3 pointed out that the stones feature a very specific form, which should be exploited for the puzzling algorithm. Independent of the discussion how the stones were set in place it is safe to assume a very high fitting accuracy and precision leading to nearly

invisible transitions between the stones. The following chapters elucidate how to make use of this unique geometric form and the tight fitting. The outline of the algorithm that has been developed based on the presented research and considerations in order to solve the Khmer temple puzzle is as follows: 1) for each digitized 3D model of a sandstone block a minimum volume box (MVB) is created (see chapter 4) and possible pairing combinations (i.e. how two stones join) are enumerated. 2) For each combination a similarity analysis is conducted, see chapter 5 and a similarity value is computed. 3) The pairing combinations are pruned according to their similarity value and the best ones are kept, see chapter 6. 4) The remaining combinations are pieced together to form part of the original wall, see chapter 6 and chapter 7. Finally, the suggested solution is validated.

Chapter 4

Simplifying the model

> *Every solution to every problem is simple. It's the distance between the two where the mystery lies.* [28]

The data that is used for the partial virtual reassembly of a broken wall from the Banteay Chhmar temple site has been gained using a high-resolution 3D structured-light and stereo vision scanner (cf. chapter 3.3, Schaefer et al. [SMF+11] and Freudenreich et al. [FSN+11]). Figure 3.13 depicts the acquisition process in short: The stone is cleared, scanned and subsequently a virtual 3D model is created. Each of these models resembles the original object with an accuracy of 3.14 points/mm^2 on average resulting in a data size ranging from 75 MB for the smallest model of a stone up to 1.1 GiB for the largest model. Since the data size increases with every stone that is added to the puzzle (see section 2.6) a meaningful way of reducing the size of each model is needed. To be able to solve this large-scale 3D puzzle, requires an algorithm that gives information on the quality of two corresponding pieces. In the current case the threedimensional shapes resembling the form of blocks are used for the puzzle, i.e. corresponding parts refers to two rectangularly shaped bounded surfaces from two different blocks. This chapter introduces minimalistic versions of the high resolution models which are used whenever the acquired high resolution is not necessarily needed, how the input data is segmented into parts that can be compared to each other and how matching candidates are determined. The minimalistic stone representations are based on so-called bounding boxes which will be explained in the first part of this chapter.

[28] Derek Landy, Skulduggery Pleasant

4.1 Bounding Volumes

Imagine a virtual surrounding in which a character can walk through the environment and explore it. The virtual figure is able to recognize structures like walls, plants or people and can interact with them without accidentally passing through them. This is enabled by intersection tests, that apply the principle of *bounding volumes*. The significant advantage of testing versus a bounding volume in contrast to the whole object is that the object itself (e.g. a person, a house, an animal, ...) is typically composed of many vertices, faces or polygons to make it as realistic as possible, thus the representation is a highly complex shape. Contrary to this a bounding volume is a simple geometric object, e.g. a sphere, that completely surrounds a given object. Instead of checking a complete polygonal model for intersection (computational amount: $O(n^2)$) only its bounding volume is tested. As long as the bounding volume remains untouched, the complex object inside does not need to be considered for further computations (this is also known as "early out" concept). For a correct and real-time decision whether there is an unwanted intersection or not easy shapes facilitate fast intersection tests. The following section is primarily based upon Bender [BB06], Eberly [Ebe07], Foley [FDF+97] and Mortenson [Mor06]. For more information the reader is referred to the aforementioned literature. Among the areas, where bounding volumes are used are:

- Computer graphics, e.g. Ray tracing: used for ray intersection and/or viewing frustum[29] tests. If the ray does not intersect with the bounding volume, it will not intersect with the object itself either.
- Computer games: used for collision detection. If the player hits a wall, door or gets hit by something it is faster to first test if the bounding volume is in collision.
- Robotics: used for collision detection to prevent the robot from unwanted collisions.
- Animation and computer simulated environments: used for collision detection and occlusion of animated objects or while interacting within a simulated environment.

Depending on the purpose and the shape of the underlying object bounding volumes can have different forms, such as 1) bounding sphere, 2) bounding box, 3) bounding ellipsoid (tighter fit than a sphere), 4) bounding cylinder (appropriate if the object can only rotate around a vertical axis) or 5) convex hull. Examples of the principle boundary volumes are shown in figure 4.1.

[29] The viewing frustum is the volume containing everything visible on the screen.

Bounding volumes taking the shape of a box can appear in two ways: as 1) AABB (axis aligned bounding box) or as 2) OBB (oriented bounding box). The first is a rectangular box, aligned with the axis of the current coordinate system completely surrounding the object in question. An AABB is very simple and fast to compute as just the minimum and maximum values of vertices in all three space directions need to be calculated. The drawback of this approach yet can be seen in fig. 4.1: An OBB is tighter fitting than an AABB, thus with an AABB intersections are more likely to lead to false positives, i.e. an intersection is reported but it is an intersection with the bounding box only and not with the object itself. In case of an OBB the bounding box is rotated such that it encloses the object in the minimal possible way leaving less free space between object and bounding volume.

The choice and size of a bounding volume are determined by computational cost, updating cost, cost of intersection detection and precision of interest. In general it can be said, that the more sophisticated the bounding volume is, the more expensive the tests get. In practice there are often several types used in conjunction.

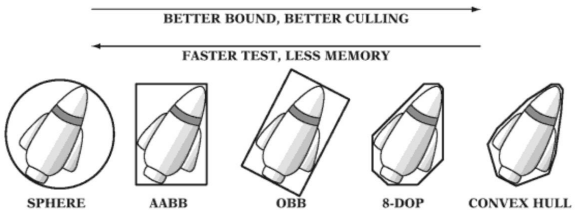


Figure 4.1: *Different types of bounding volumes. Picture source: Ericson [Eri05]*

The characteristics of a bounding volume are 1) tight fitting, meaning the volume resembles the real object as closely as possible, 2) intersection efficient, meaning it is fast and easy to work with and 3) memory efficient, meaning they require only a small amount of memory. The advantage is that bounding volumes can be generated during compile time not during run-time making them fast accessible. One of the disadvantages however is that if the object is transformed, the bounding volume needs to be transformed as well. The smallest amount of memory is needed for a sphere, as only the center and the radius have to be stored.

The following sections will illustrate the correlation between bounding volumes and Khmer temple stones and describe the workflow necessary to compute a volume tightly fitting to a virtual temple stone. Existing approaches to automatically compute a tight fitting volume box could not be applied to digital Khmer temple stones (see section 2.3) as their major drawback was the difficulty to automatically detect whether a temple stone

features an indentation or not. Therefore a new representation called *minimal volume box* tailored to Khmer sand stone blocks as well as a method to derive such a box is developed. In the first step of the method an OBB for each stone model is computed, then edges (i.e. a neighbouring pair of connected vertices, see next section) of the stone that correspond to the edges of the minimal volume box are extracted, secondly it is checked, whether an indentation is present and thirdly, the box gets refined. Those steps are applied to all stone models. In preparation for the computation of matching parts needed to solve the puzzle, each box is classified whether having no indentation, one indentation or two indentations.

4.2 Bounding Boxes for Khmer temple stones

In general, before a puzzle can be solved a method is required classifying matching surfaces and judging how closely two sides resemble each other. Furthermore this method needs to be capable to quickly distinguish between different sides and assess which side is to be used for a potential match. In case of temple stones matching surfaces are the vertical and horizontal joints. Thus the virtual 3D model needs to be segmented and those joints determined. The difficulty of an automatic segmentation are broken corners and eroded edges. The bas-relief most stones feature should not be used in the puzzling algorithm as in some cases there has not been a bas-relief in the first place and in some cases it has gone missing due to weathering or looting. Drawing the attention to the last passage of the preceding subsection, it was conducted that bounding boxes are tight fitting, intersection efficient and memory efficient. The oriented bounding box of a Khmer temple stone, see figure 4.2 already closely resembles the stone.

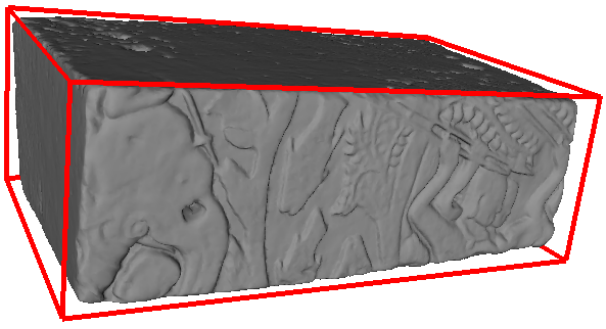


Figure 4.2: *A 3D model of a Khmer temple stone and its oriented bounding box.*

Furthermore as the surfaces of each stone are approximately planar the high-resolution models can be reduced to drastically simplified versions using scientific methods to decrease computational amount and time later on. It has to be stressed that an investigation

on the resemblance of two joints cannot be confined to the planar sides of a bounding box as important curvature information would be lost and the similarity proposition worthless. To avoid ambiguity the following definitions are made which are based on Ball, [BC87].

Definition: Vertex

A *vertex* v is defined as a point in 3D space $v := \{p \in \mathbb{R}^3 | (p_x, p_y, p_z)\}$. A vertex can either be the corner point of a polygon, the ending point of a line or just a point.

Definition: Polygonal line, polygon

For a set of vertices $\{p_0, ...p_n\}$ the set $Q := \{(p_0, p_1), (p_1, p_2), ..., (p_{n-2}, p_{n-1})\}$ is called a *polygonal line*. The pairs of vertices are the *edges* of the polygonal line. If $p_{n-1} = p_0$ it is called a *closed polygonal line*. The area that is surrounded by a closed polygonal line is called a *polygon*.

Definition: Face

If a polygon is part of a greater object which is constructed from several, not necessarily identical, polygons (e.g. cube), it is called a *face*.

Definition: Polyhedron

A set of polygons will be called a *polyhedron*, if the following holds:

(i) Each two faces share either no vertex or one corner vertex or one edge. The intersecting set of two different faces is therefore either empty, a corner vertex or an edge.

(ii) Each edge of a face belongs either to one or at most two faces.

(iii) The set of all edges belonging to only one face is either empty or forms a closed polygonal line. This polygonal line is called the *boundary* of the polyhedron.

Definition: Box faces

An oriented bounding box has got six faces and is assumed to be oriented parallel to the coordinate axis of a cartesian coordinate system and that one vertex lies in the origin of the coordinate system. Then, the six faces will be termed

- *Front:* the face with the lowest overall z value.
- *Back:* the face with the highest overall z value.
- *Left:* the face with the lowest overall x value.
- *Right:* the face with the highest overall x value.
- *Bottom:* the face with the lowest overall y value.
- *Top:* the face with the highest overall y value.

The stones are the pieces of the 3D puzzle and thus a tool describing fitting and non-fitting parts is needed. As the temple stones feature a box-shaped form the preferred fitting criterion is how the faces of this box match together. A box-shaped form based on an OBB suits ideally for fast and reliable matching tests, segmentation of the high-resolution 3D models and validation of the suggested solution. Referring to the specific geometric form of a Khmer temple stone, the most typical feature is an indentation, which some of the stones incorporate. Yet, between an OBB and an indentation a lot of free space is retained which indicates to prefer a tighter fitting volume. The algorithm involved in computing such a volume must be able to detect indentations automatically, categorize and construct a tight fitting box to the stone. As the edges of the 3D model coincide with the edges of such a box, the first stage of the algorithm is to perform an edge extraction.

4.3 Computing the boxes' edges

The virtual 3D model of temple sandstone blocks can be represented as either point clouds or as triangulated meshes depending upon the requirements. In order to compute a minimal tight fitting box it is necessary to distinguish faces and edges. Mathematically, edges are areas featuring high curvatures, i.e. a curvature computation is needed. In this

thesis curvatures are computed based on *algebraic point set surfaces (APSS)* as they provide a reliable and computationally cheap estimate of the mean curvature of a surface and allow robust handling of sharp features and boundaries based on moving least squares, see Guennebaud [GG07]. Originally, *point set surfaces (PSS)* were developed by Alexa et al. [ABCO+03] for fast and facile display of 3D models whose overall point density is rather low. This approach is able to dependably compute additional points and thus enhance the models visual representation. APSS's create algebraic spheres for the point sampling thus the curvature of the underlying mesh is already given as the ratio of $\frac{1}{r}$, where r is the radius of the sphere. The following description is based on the papers from Guennebaud et al. [GG07], [GGG08] and outlines the APSS approach:

A given point set $P = \{p_i \in \mathbb{R}^d\}$ should be interpolated. This is done by computing the implicit scalarfield $f(x)$ given in equation (4.1), which depicts the algebraic distance between the evaluation point x and the sphere $u(x)$.

$$f(x) = S_{u(x)}(x) = [1, x^T, x^T x]u(x) = 0, \tag{4.1}$$

where $S_{u(x)}(x) = [1, x^T, x^T x]u$, $u = [u_0, ..., u_{d+1}]^T \in \mathbb{R}^{d+2}$. u is a scalar vector describing the sphere. The center of the sphere is given as:

$$c = -\frac{1}{2u_{d+1}}[u_1, ..., u_d], u_{d+1} \neq 0 \tag{4.2}$$

and the radius is

$$r = \sqrt{c^T c - \frac{u_0}{u_{d+1}}}. \tag{4.3}$$

The sphere fit for a given point x is then:

$$u(x) = arg \min_{u, u \neq 0} \left\| W^{\frac{1}{2}}(x) \cdot D \cdot u \right\|^2, \tag{4.4}$$

with W being the weighting matrix

$$W = \begin{pmatrix} w_0(x) & & \\ & \ddots & \\ & & w_{n-1}(x) \end{pmatrix}, \tag{4.5}$$

and the functions $w_i(x)$ and ϕ are given as

$$w_i(x) = \phi\left(\frac{\|p_i - x\|}{h_i(x)}\right), \quad \phi = \begin{cases} (1 - x^2)^4, & x < 1 \\ 0, & \text{else.} \end{cases} \tag{4.6}$$

D is given as

$$D = \begin{pmatrix} 1 & p_0^T & p_0^T p_0 \\ \vdots & \vdots & \vdots \\ 1 & p_{n-1}^T & p_{n-1}^T p_{n-1} \end{pmatrix} \tag{4.7}$$

Applying several transformations (see Guennebaud et al. [GG07], [GGG08]), the algebraic sphere fit can be given as

$$u(x) = (D^T W(x) D)^{-1} \cdot D^T W(x) \cdot b = A^{-1}(x) \cdot \hat{b}(x), \tag{4.8}$$

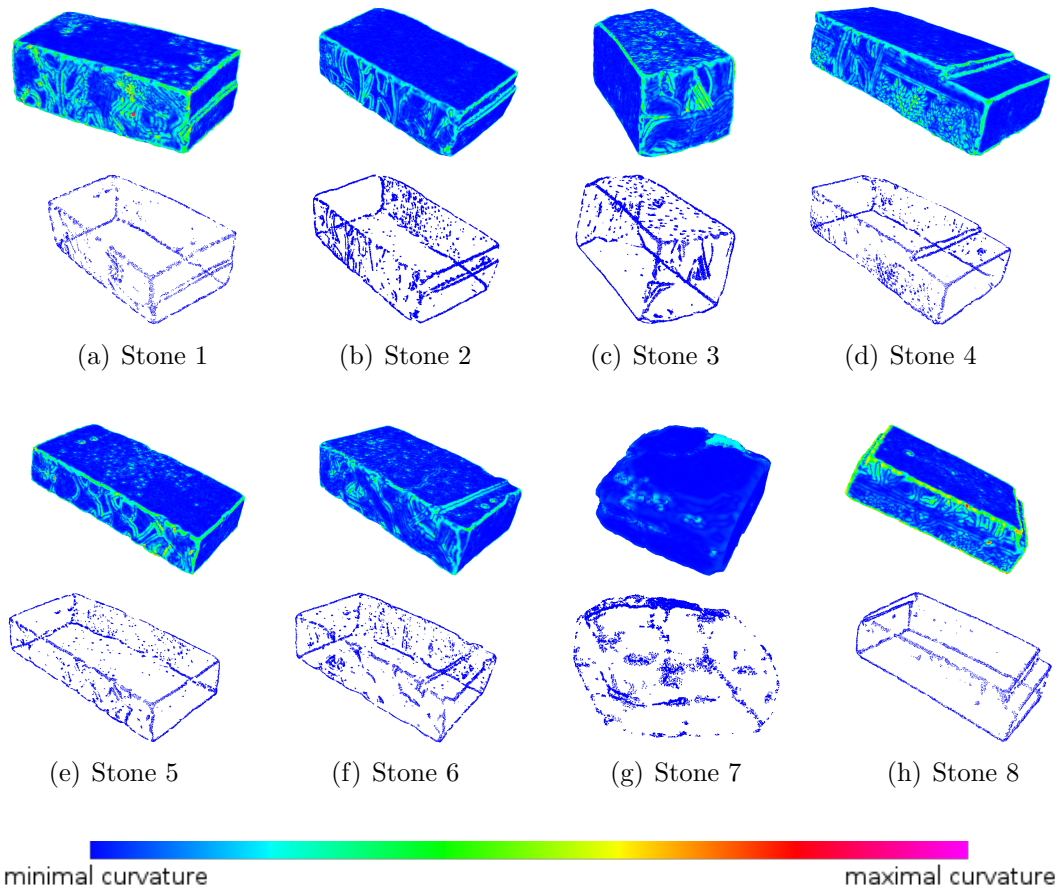
with

$$W = \begin{pmatrix} w_0(x) & & & & \\ & \ddots & & & \\ & & \beta w_i(x) & & \\ & & & \ddots & \\ & & & & w_{n-1}(x) \end{pmatrix}, D = \begin{pmatrix} \vdots & \vdots & \vdots \\ 1 & p_i^T & p_i^T p_i \\ 0 & e_0^T & 2e_0^T p_i \\ \vdots & \vdots & \vdots \\ 0 & e_{d-1}^T & 2e_{d-1}^T p_i \\ \vdots & \vdots & \vdots \\ 1 & p_{n-1}^T & p_{n-1}^T p_{n-1} \end{pmatrix}, \tag{4.9}$$

$$b = \begin{pmatrix} 0 \\ \vdots \\ e_0^T n_i \\ \vdots \\ e_{d-1}^T n_i \\ \vdots \end{pmatrix}. \tag{4.10}$$

β is an additional weighting factor that is usually set to $\beta = 10^8$, e_i is the unit vector and n_i the unit normal.

Figure 4.3 shows the result of this APSS based curvature estimation applied to the 3D models of Khmer temple stones. The color code ranges from blue for areas with the lowest curvatures over cyan and green to red for areas with the highest curvatures. It can clearly be seen, that areas with a higher curvature are located primarily on the edges of the model. The resulting point clouds differ depending on the underlying stone (e.g. whether its side faces have a lot of chisel marks or a deeply incised bas-relief), the quality of the virtual model, the radius of the sphere and the threshold value used for the extraction of highly curved areas.



(a) Stone 1 (b) Stone 2 (c) Stone 3 (d) Stone 4

(e) Stone 5 (f) Stone 6 (g) Stone 7 (h) Stone 8

Figure 4.3: *Eight of the digitally acquired stones with their estimated curvatures and the extracted point cloud containing highest curvatures only. It can clearly be seen that areas with higher curvatures generally correspond to the stones' edges. Exceptions are the bas-relief on the front or the unhewn back side.*

4.4 Detection of indentations

With the extracted point cloud containing points from areas with high curvatures only the succeeding procedure to compute a tight fitting volume is an automatic detection of whether indentations are present or not. For a human this is a very straightforward task as the indentation can be effortlessly recognized in figure 4.3 (a), (b), (d), (f) and (h). In contrast to this an algorithm is not competent to distinguish points belonging to an indentation or to any other edge and therefore cannot conduct if an indentation is present without relying on further information. To this end, several approaches were investigated. For the following discussion it is assumed that the extracted point cloud is oriented such that an indentation is only found on the top left or top right side. The *length* of a stone is taken as the distance from left face to right face, the *height* as the distance from bottom face to top face and the *width* as the distance between the front face and the back face.

Several techniques were tested to find a way to automatically detect indentations. The first approach was to use a plane that 'travels' the width, length and height of the stones' extracted point cloud and counts the total number of points it passes. If an indentation is present the number of counted points should be significantly higher than in cases where no indentation is present. Figure 4.4 shows that this assumption did not work. The red line depicts the number of counted points for the stones 1, 3, 7 and 8 (see figure 4.3). Theoretically, the red line should at some point show a rapid increase in counted points for stones 1 and 8, but obviously no clear distinction between the four cases can be observed.

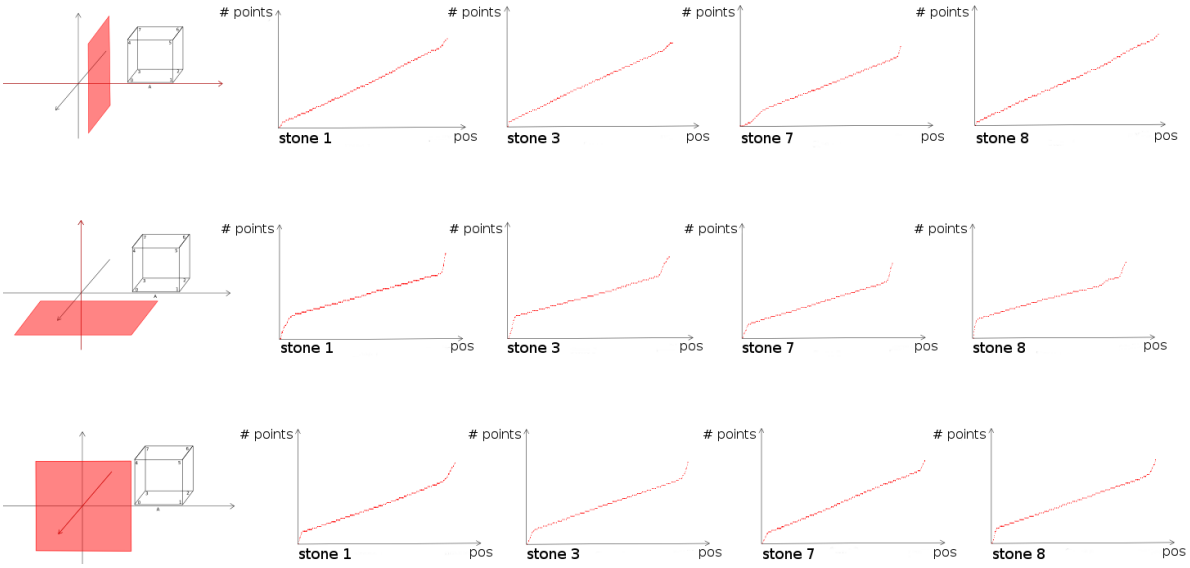
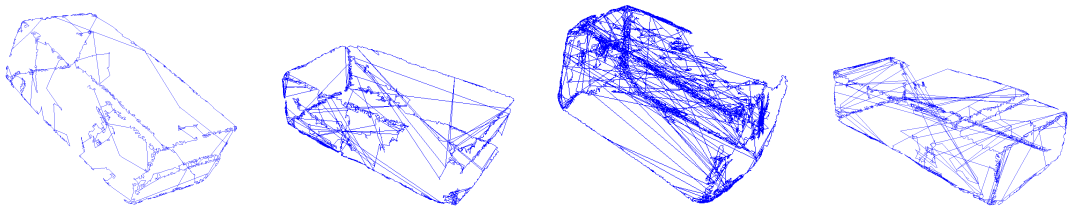


Figure 4.4: *Taking a plane in 3D space that 'travels' the width, length and height of an extracted point cloud counting the number of points it passes, it is assumed that a significant difference between models with an indentation (stone 1, stone 8) and models without an indentation (stone 3, stone 7) would occur.*

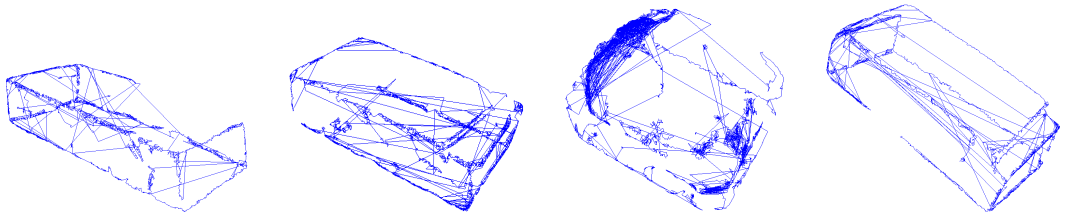
To understand the failure of this approach, the extracted point clouds, shown in figure 4.3 need to be observed. These contain so-called noise meaning there exist areas with a high curvature that do not belong to the edges but to the bas-relief on the front side, the back side, holes or deep chisel marks. These points falsify the result and make the method unsuitable.

The next approach was to use a *minimum spanning tree*. In the research field of graph theory a spanning tree is considered to be a connected, undirected subgraph of G, including all nodes and a minimum of its edges. A *minimum spanning tree (MST)* is a spanning tree whose overall weight is less than the weight of every other spanning tree, for further information see chapter 6. Taking the distance between the points as weights, the assumption is that junctions (i.e. a point being connected to more than two other points) in this MST would predominantly occur at the corners of the stone model. Figure

4.5 shows the minimum spanning tree algorithm applied to eight stones. It can be seen, that the method can not cope with the noise as well and therefore junctions occur not only on the edges but on various areas of the 3D model.



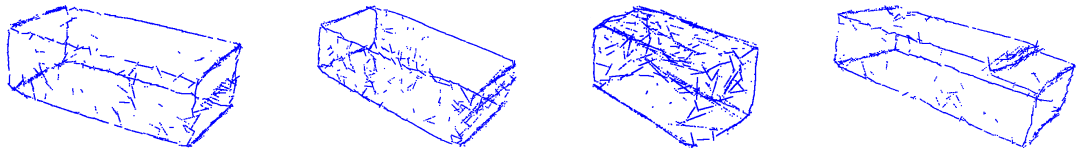
(a) Stone 1 (b) Stone 2 (c) Stone 3 (d) Stone 4



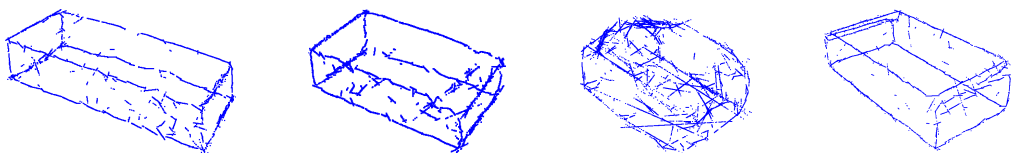
(e) Stone 5 (f) Stone 6 (g) Stone 7 (h) Stone 8

Figure 4.5: *A minimum spanning tree algorithm has been applied to the curvature points to automatically detect the edges and cornerpoints.*

The next method applied to the extracted point clouds was the detection of sharp features based on the works of Daniels II et al. (see chapter 2). The authors suggest to smooth the extracted points, project them onto a polyline and extract line features. Figure 4.6 shows the result for Khmer temple stones. In case of clear edges as shown in the papers from Daniels II et al. [DIHOS07], the point cloud is able to be transformed into a complete wireframe model. In contrast to that, for the virtual stone models the polylines get strongly blurred and it is not possible to automatically detect, whether there is an indentation or not.



(a) Stone 1 (b) Stone 2 (c) Stone 3 (d) Stone 4



(e) Stone 5 (f) Stone 6 (g) Stone 7 (h) Stone 8

Figure 4.6: *Daniels II. et al. model applied to Khmer temple stones leads to a blurring of the indentations, most prominent at stone 1 and stone 2.*

None of the investigated methods is able to distinguish between an edge line and an indentation line although all of them are capable of detecting line structures inside a point cloud. As several approaches to detect indentationes automatically failed, a semi-automatic semi-interactive procedure was developed to calculate the MVB of each temple stone.

4.5 Computation of tight fitting bounding boxes for Khmer temple stones

The first step of the herewith introduced semi-automatic semi-interactive method to simplify the high-resolution 3D stone models into a tight fitting volume edge-based model is an OBB computation and a curvature estimation for each stone. Using the vertices of the OBB the user corrects them in the second step in case the underlying 3D temple stone has broken parts or adjoins additional vertices in case an indentation is present. These improved vertices are used as a basis for the automatic corner vertex computation in the third step, where the updated vertices are refined. For each pair of neighbouring vertices a cylinder is laid out whose height is given as the euclidean distance between the two vertices and whose best radius for all stones was in practice determined as $r = 15.0$ cm. All vertices from areas with a high curvature, lying inside this cylinder, are fitted to a line using a linear least squares approach. Each three (as it is a problem in $\mathbb{R}^3$) of those skew lines meet in one of the corners of the new box, thus the corner vertex is obtained by computing the vertex having the minimal distance to all three lines. From a mathematical point of view there is no minimal distance between three points in $\mathbb{R}^3$. Therefore the practical approach was to determine which two lines feature the minimal distance and choose a vertex which bisects this distance. Figure 4.7 shows an overview to this simplification process. Th new kind of representation for Khmer temple stones is termed *Minimal Volum Box (MVB)*.

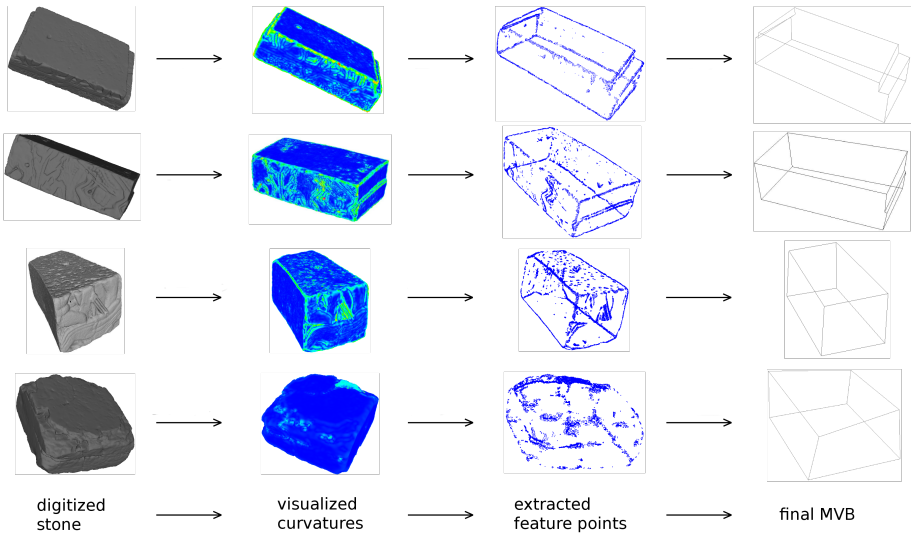


Figure 4.7: *The simplification process from 3D modell to wireframe modell is shown for stone 1, stone 3, stone 7 and stone 8. A curvature estimation based on APSS is applied to the digitally acquired stone and areas with high curvatures extracted. By refining the vertices of an OBB of each modell and using a cylinder test, the final tight fitting volume box (MVB) is computed.*

Definition: Minimal Volume Box

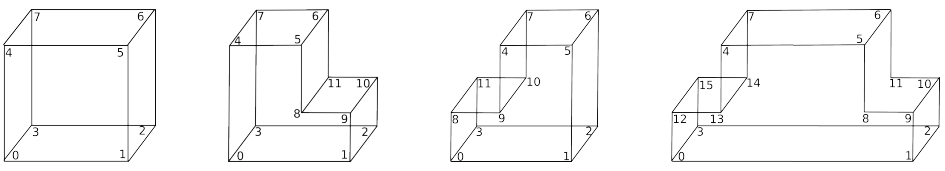
The minimal volume that represents a simplified digital Khmer temple stone satisfies the requirements of a polyhedron. Moreover, as every angle between two edges is in between 80° and 100° it resembles the form of a cube or a cube with indentations respectively. The here studied MVB's show the following systematics:

a) 8 corner vertices and 6 faces

b) 12 corner vertices, 8 faces and an upper right indentation,

c) 12 corner vertices, 8 faces and an upper left indentation,

d) 16 corner vertices, 10 faces and an upper indentation on the left and on the right.



As discussed in chapter 3 the temple site of Banteay Chhmar features many similarities with the better known Bayon temple and, in general, with many other Khmer temples from that period of time. In conclusion the pricinple of a MVB can be used for the reassembly not only of parts of Banteay Chhmar but also of other temple sites in former Siam. As the concept of a MVB is very flexible it is moreover possible to extend it to other types of stones, e.g. stone blocks used for arch or tower building.

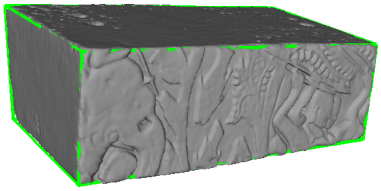


Figure 4.8: *A stone and its MVB (shown in green) showing the MVB is an inner approximation.*

Comparing the volume of the OBB of a virtual Khmer temple stone with the volume of its MVB the latter one is an underestimate whereas the former approach is an overestimate. Therefore the oriented bounding box completely encloses the 3D model of a sand stone block, whereas the minimal volume box resembles it. Due to the linear least squares method which was applied to the high curvature areas to create the MVB its edges are not an outer approximation but an inner approximation. Figure 4.9 shows the calculated deviations in percent of the volumes of all stones computed for the OBB's (blue) and the MVB's (red) and as well as original volume as a green reference base line. Using an OBB the volume is overestimated on average by 23.58% while with the MVB the volume is underestimated by 10.43% on average. The overall percentaged error is lower in case of the MVB.

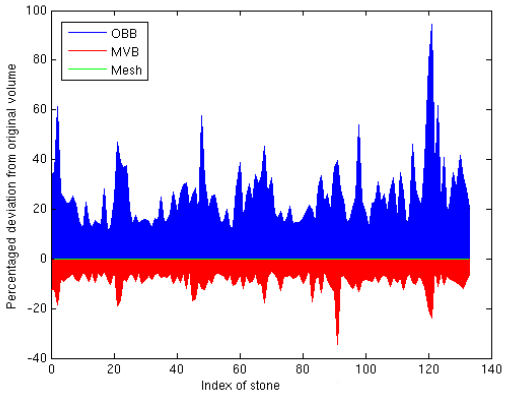


Figure 4.9: *The deviation in percent from the volume of the original 3D stone model is shown for all acquired stones in case of an OBB (blue) and in case of a MVB (red). The green line depicts the original volume as a reference base being 100%.*

In figure 4.10 the distribution of volumes for all digital stone models is shown. The distribution was determined using a kernel density estimation function. It gets clear that the distribution of MVB volumes closely mirrors the original volumes.

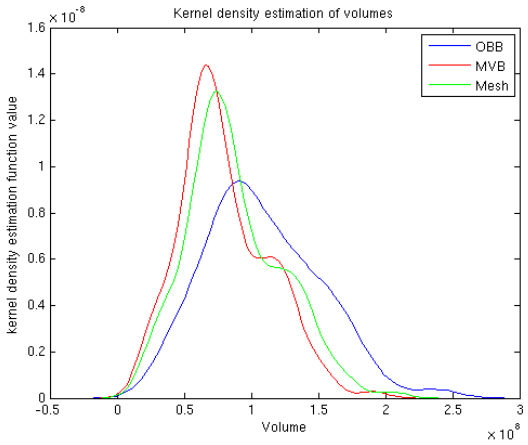


Figure 4.10: *The distribution of original volumes (green line), volumes of MVB (red line) and OBB volumes (blue line) of all digital stones. The MVB closely resemble the distribution of the volumes of digital Khmer temple stones.*

Classification and segmentation

For the match detection and similarity analysis of Khmer temple stones in general and for the developed puzzle algorithm in particular it is important to know whether the stone model (and its MVB) have one or more indentations and on which side of the stones these are located. The studied stones can have up to two indentations which are located either left or right. Using this knowledge, the MVBs get classified to be able to distinguish between logically meaningful and impossible combinations of two virtual stones. Figure 4.11 shows how the classification step is performed using stone 1 as an example. As it is not possible to automatically detect whether an indentation exists or not each 3D model is tagged with this information when the corner vertices are refined. Therefore this information can be extracted from the 3D model without further calculations.

An additional information which is written into the 3D model during the digitization step is its orientation. Stone conservators and craftsmen on site already know the orientation of each stone which is why this information does not need to be time-consumingly computed. Combining the MVB its classification and its orientation, the virtual three dimensional sand stone blocks get segmented into their left, right, top, bottom and indentation parts.

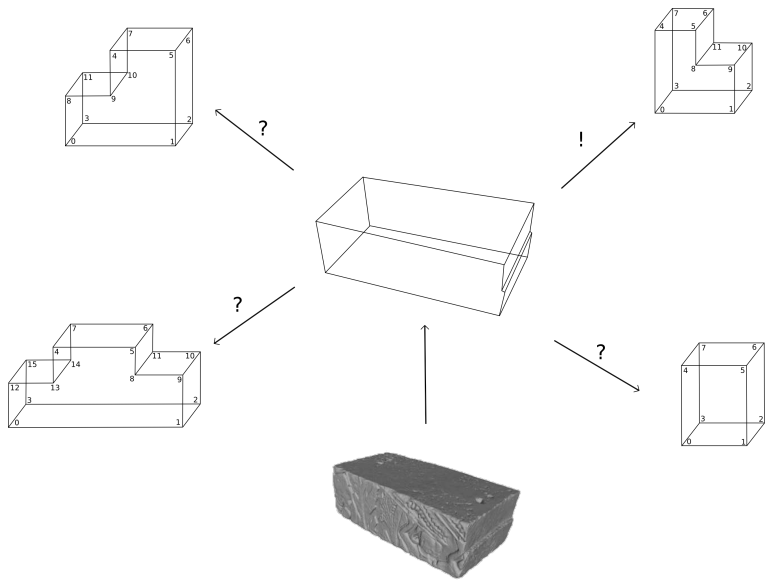
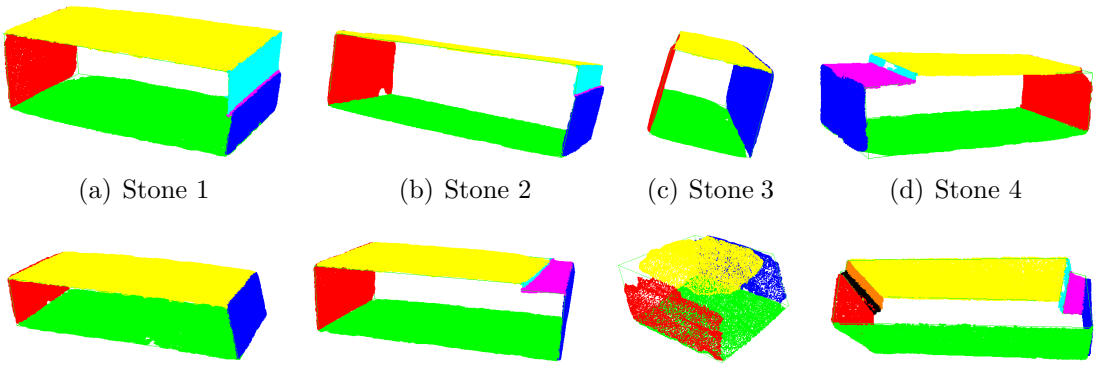


Figure 4.11: *After the wireframe model of a stone has been determined, it is categorized.*

The front side and the back side are not considered for further measurements and evaluation. The back side usually does not contain useful information (except for stones from the top most or bottom most row). Extracting information about the bas-relief can provide helpful information wether two stones fit together. Yet as not all of the stones contain a bas-relief, some by accident some on purpose, it was not considered in this thesis.

The segmentation of the digital stone model into left, right, top, bottom and indentation is performed by extracting all vertices whose Euclidian distance is within a small threshold from their corresponding MVB face. Figure 4.12 illustrates the result of the segmentation step which is afterwards used for the similarity analysis.



(a) Stone 1 (b) Stone 2 (c) Stone 3 (d) Stone 4

(e) Stone 5 (f) Stone 6 (g) Stone 7 (h) Stone 8

Figure 4.12: *Eight of the digitally acquired stones with their segmented side faces.*

4.6 Possible pairwise combinations

Solving a 3D puzzle implies knowledge about how pieces could possibly fit as not every potential way to bring together two parts is necessarily the correct solution. In the current case a possible match between two stones is a vertical or horizontal joint of two faces, i.e. left, right, top, bottom and indentation faces. The thickness of the wall which serves as a groundtruth in the current case is determined by the width of the stones, as no stones were placed behind each other. In conclusion a front face and a back face do not serve as valid joints. Pointed out in section 4.5, the orientation of a stone is known a priori which excludes further possibilities such as combining a left face with a front face. This section presents the combinatory potentiality of how two stones can be joint in order to evaluate their resemblance using the similarity criterion explained in chapter 5. This is done for the case of two stones with no indentations, a complete enumeration is accordingly done in Appendix A.

It is common knowledge by architects, stones conservators and other experts working with Khmer temples that the construction of such temples is distinct with regard to the placement of the stones, i.e. there is one exact placement for each stone. Combining this with the fact that the stones join very tightly it is safe to concentrate the possibilities which need to be tested for the similarity analysis on flush edges. Even if a stone lies in the middle of another stone it has at least one shared edge with some other stone. This is shown by figure 4.13. It is an excerpt of figure 3.12 where the focus is laid on flush edges. The red arrows depict edges flush with edges of a neighbouring stone.

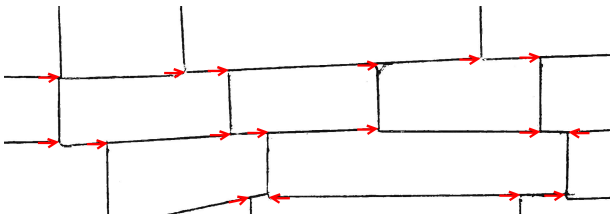


Figure 4.13: *Due to the unique placement of the stones and their tight joints it is safe to concentrate the enumeration of possible pairings on flush edges. Every stone has at least one shared edge with another one. The red arrows in this figure illustrate such flush edges.*

In general there are three ways of piecing the stones together to solve this problem: First, starting with pairwise combinations increase matches to larger joining parts until the solution exists as one piece (this is also known as hierarchical clustering with a computational cost of $O(n^3)$). Second, find all pairwise combinations, evaluate them and set them together (computational cost $O(n^2)$). Third, do a combination of first and second. The significant advantage of the second approach is that the similarity computation only

needs to be performed once on comparatatively small puzzle pieces whereas for the first and third approach the similarity analysis needs to be undertaken for all pieces, including the newly emerging larger ones.

Due to the fact that each stone can be classified to fit into one of four different types an enumeration of all valid pairwise combinations can be performed beforehand, thus saving time. As stones have at least one edge flush, the orientation is known and false pairings can be excluded, the remaining enumeration is done based on edges resembling the width of the wall (in the current case being the width of the stones). Figure 4.14 illustrates eight of them. For the similarity analysis later only the surfaces being in contact with each other will be considered (marked in red in figure 4.14). The complete enumeration and storage of all possible solution pairs is done using the MVB.

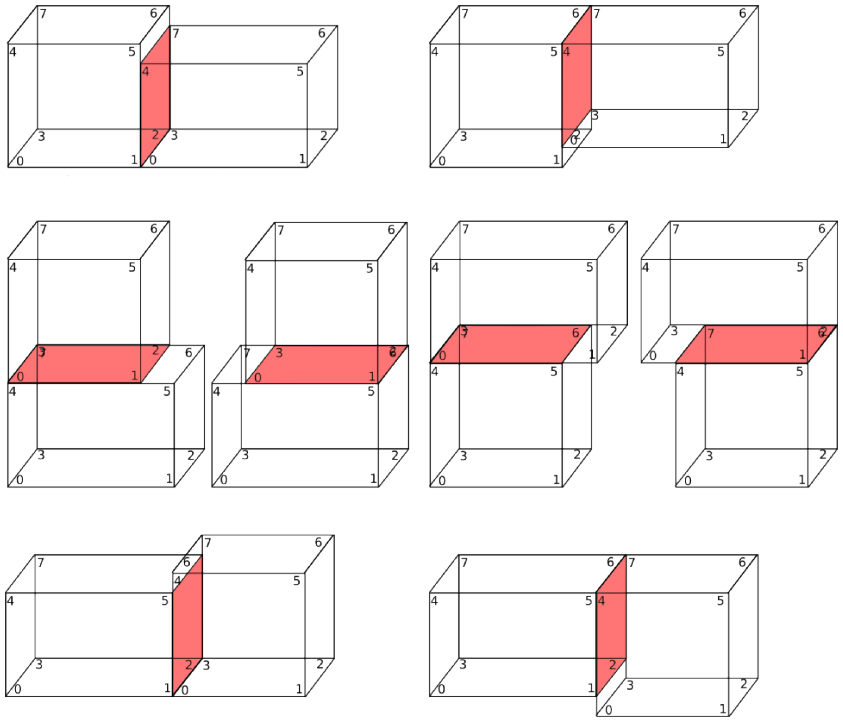


Figure 4.14: *Possibilities on how to combine two MVBs with eight corners respectively. The edges corresponding to the width of the stone are taken as edge of origin and the surfaces in contact will be considered for further similarity analysis.*

Chapter 5

Similarity analysis

> *Similarity is fundamental for learning, knowledge and thought, for only our sense of similarity allows us to order things into kinds so that these can function as stimulus meanings.*[30]

Chapter 3 gave an overview on the building techniques of Khmer temples and in chapter 4 a new method of computing and classifying a simplified model of the digital stone models as well as possible pairwise matches were presented. The next step of the puzzle pipeline is a similarity analysis making reliable statements on the quality of a possible match, thereby giving room to filter better and worse fitting pairs or even find the correct solution between all possible solutions.

The general question *"What is similarity?"* is not easy to answer as the subject of similarity is dealt with in a wide range of disciplines ranging from psychology over physics to mathematics. The Oxford Handbook of Thinking and Reasoning states that *"Humans and other animals perceive and act on the basis of similarities among things because similarities are usually informative. Similar things usually behave similarly, and because we can grasp these similarities, we can organize and predict the things in our world."*[31] It suggests four classes of distinction: geometric models (similarity is based on position in multidimensional space), featural models (similarity is determined by the number of shared features), alignment-based models (similarity is measured through corresponding structures) and transformational models (number of transformations needed to convert

[30] Oxford Handbook of Thinking and Reasoning; p. 155
[31] Oxford Handbook of Thinking and Reasoning, p. 155

one model into the other). Distinction through transformational models is also called *shape similarity* and is defined as *'we are given two objects A, B and want to know how much they resemble each other. Usually one of the objects may undergo certain transformations like translations, rotations or scalings in order to be matched with the other as well as possible.* [32] Concerning the case of Khmer temple stones no additional information can be learned from this approach as the stones already closely resemble each other. For the same reason a topological similarity analysis cannot be applied, as the stone models are topological equivalent. Featural models require the virtual models to be described according to different but comparable features, such as existence of an indentation, or a bas-relief or the width of a stone. Yet, such features are not distinct enough to make reliable statements to reassemble the wall. An alignment-based model for similarity is not sufficient as a corresponding structure in the current case would mean to check which side faces of a stone could possibly fit. What is needed for the solution of the puzzle is an analysis of how good in the sense of geometrical closeness such an alignment can be compared to another alignment. This leads to similarity based on geometric models.

The virtual 3D models of the Khmer temple stones consist of triangulated discrete point clouds. To gauge similarity in terms of positions in space making use of the minimal volume boxes from the previous chapter can be done by measuring the angle of intersection between the associated faces of two MVB's. In the present case, though, the results are nondiscriminable from each other as it is possible for two boxes to have an ideal angle of $0°$ and not belong to each other. The temple stones were acquired using a high-resolution 3D scanner with an accuracy of 3.14 points/mm^2. Therefore the inter-point distance (= distance between two points belonging to one stone) is lower than the expected distance between the side faces of two stones belonging together. This led to the decision to develop a similarity criterion based on points. All methods which are taken into account for this purpose are evaluated at each point of one stone face possibly being in contact with a point from a corresponding stone face.

The following presents and discusses several point-based methods to compare two models with each other and derives a similarity measure for Khmer temple stones. To strengthen the similarity analysis and allow for reliable decisions a combination of different per-point measures is used. Those measures use two-dimensional and three-dimensional joining characteristics. Usually, correct matchings have a high score in all measurements. To clarify the meaning the following annotations are made according to [SBSL14].

[32] A. Adan and M. Adan. A flexible similarity measure for 3d shape recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11):1507–1520, November 2004., p. 1

- A **matching** or **pairing** $p(i, s_i, j, t_j)$ is a pair of two stones combined at either one side surface. i, j denotes the stones and s_k, t_k denote the faces of the stones being in contact with each other.
- A **correct matching** or **correct pairing** is a pair of two stones combined such that an expert can verify their proper mutual arrangement.
- A **solution** is a valid join of several pairings. If there are n stones in the examined dataset, a **complete solution** consists of $n - 1$ pairings.
- A **correct solution** is a solution consisting of correct matchings only.

Furthermore, for $n = 135$ being the number of digitally acquired stones, a *dataset* is termed a subset of the complete digitally acquired wall containing m stones, where $1 \leq m \leq n$.

5.1 Aligning stones for further evaluation

In 1992 Paul Besl and Neil McKay introduced the well-known and widespread *iterative closest point (ICP)* algorithm for the registration of two shapes (data and model) in 3D space [BM92]. It is not only suitable to align two shapes with each other but can also be used to make statements on the quality of this match. It was therefore integrated into the puzzle workflow in order to align two stones prior to the performance of the similarity analysis and its resulting function value is used as one of the similarity criteria. The heart of the algorithm is the following optimization problem (see equation 5.1), which is solved during each step. In this formula q is the reference model and p the target model. R with its rotational angles κ, ω, ϕ is the 3D rotational matrix, $t := t_x, t_y, t_z$ the translational vector. N is the number of elements.

$$\min_{\kappa,\phi,\omega,t_x,t_y,t_z} \quad f = \frac{1}{N} \sum_{i=1}^{N} \|R \cdot p_i + t - q_i\|_2^2 \tag{5.1}$$

The algorithm works as follows (the enumeration is closely related to the originally given one, see [BM92]):

- Select the point sets P with N_p points $\{p_i\}$ from the data shape and the model shape X with N_x points, where $N_x = N_p$ is required
- Initialize all values and start the iteration
    1. Compute closest points (p_i, q_i).

2. Compute the registration: solve optimization problem (5.1).
3. Apply the registration to all points of P
4. Terminate the iteration when the change in mean-square error is below a user-defined threshold.

- Repeat steps 1 - 4 until the termination criterion τ is met.

Due to the fact that the ICP is a very powerful and reliable method especially in the fast growing area of registering 3D surfaces many advancements in all steps of the workflow have been proposed since its introduction. Rusinkiewicz and Levoy [RL01] give a review classifying, presenting and discussing those improvements according to six classes affecting different steps of the algorithm: 1) Selection of points, 2) Matching of selected points, 3) Weighting the corresponding point pairs, 4) Rejection of certain pairs, 5) Assignment of an error metric based on the pairs, and 6) Minimizing the error metric. In the present case of aligning Khmer temple stones the best results were gained by accelerating the closest point computation using a k-d tree (a space partitioning data structure) and rejecting pairs with largest point-to-point distances.

Collision free registration

In the current case where the shapes to be matched are the sides of a stone it is important that the ideal registration does mean that the stones do not intersect. Following the idea of Flöry [Flo10] the second step of the inner ICP loop (minimizing the least squares error function thus finding the optimal rotation and translation) is extended by a constraint and rewritten into a standard *non-linear programm (NLP)*. The minimization problem formulation is now given as:

$$\begin{aligned} \min_{\kappa,\phi,\omega,t_x,t_y,t_z} f &= \frac{1}{N}\sum_{i=1}^{N} \|R \cdot p_i + t - q_i\|_2^2 \\ \text{s.t. } n_i \cdot (R \cdot p_i + t - x_i) &> 0, \quad \forall i, \end{aligned} \tag{5.2}$$

where n_i are the normals of each x_i.
Sequential quadratic programming (SQP) is the most successful and state of the art method for solving NLPs. It is not a standalone algorithm but a concept from which a variety of methods has evolved. At each iteration step a quadratic function approximates the objective function. The constraint conditions are approximated by linear functions. This is based on the principle of replacing a difficult problem by an easier-to-solve approximation. A short introduction to SQP methods, based on Nocedal and Wright [NW06],

Boggs and Tolle [BT95] and Schäfer [Sch09], is given. The main principle of an SQP algorithm is a series of iterates

$$x^{k+1} = x^k + t^k \triangle x^k \tag{5.3}$$

which converge to the optimal solution x^*. Within the method x^k is the current solution approximation value, $t \in \mathbb{R}$ the step size and $\triangle x^k \in \mathbb{R}^n$ the search direction. SQP algorithms can in general be divided into *active set* methods and *interior point* methods. The former is more suitable for small-scale problems, whereas the latter is better suited in case of large-scale problems. If a constraint equals zero it is said to be active. Active set methods therefore emphasize on active constraints, making them always feasible and they resemble the well-known Simplex algorithm for solving linear programs [Flo10]. Interior point methods search for the solution by traversing the interior of the feasible set.

In the current case the ICP achieves, as already mentioned, two things: Firstly, the stones are correctly aligned. Figure 5.1 shows this exemplarily. The left part shows the stones that were in a rough alignment using the MVBs before the ICP has been applied and the right part shows those stones in correct and perfect alignment after the application of the ICP. Secondly the final ICP function value serves as a quality measure concerning the matching of two stones. In case of a wrong pairing the result of the ICP does still look as if it would fit, yet the functional value is usually worse than for the correct case, which is illustrated by figure 5.2.

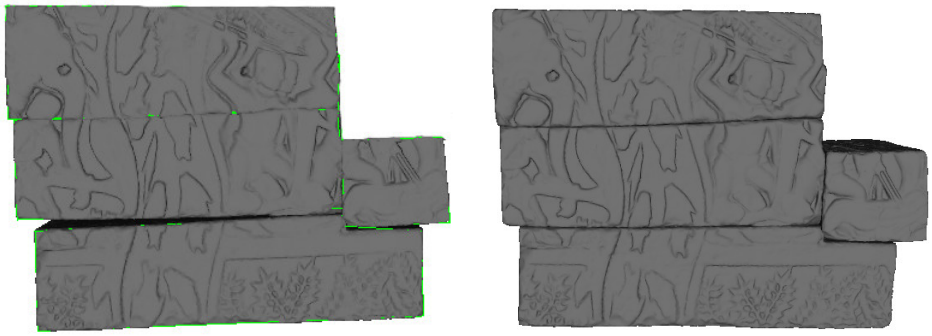


Figure 5.1: *The picture shows several stones before (left) and after (right) the ICP has been applied.*

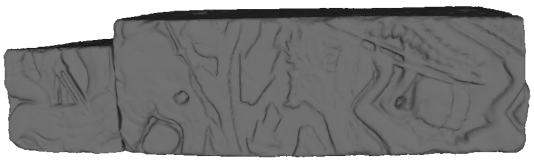


Figure 5.2: *In case of a wrong pairing the result of the ICP still seems valid, yet the functional value is in general worse than for the correct case.*

Aligning two Khmer temple stones using the ICP algorithm the computation can be stopped after ten iterations in each case. This is due to the fact that then either the accuracy does not significantly improve any further or that the algorithm finds another local minimum wich minimizes the function but does not correctly align the point clouds in question. Depending on the number of points taken into account and the quality of the underlying mesh[33] the final function value of the ICP lies on different scales. It is possible that $minf = 5.1$ or $minf = 7.0$ or $minf = 2.99$ for correct cases. This hinders an efficient comparison of ICP values and was one of the reasons to combine several similarity measurements. The explanatory power for different matchings is checked by using six selected datasets, see figure 5.3. The datasets are chosen to differ in number of stones included as well as in quality of their measurement. In the following, dataset 01 refers to the stones marked in green in figure 5.3, dataset 02 to the stones marked red, dataset 03 refers to the blue stones, dataset 04 to the yellow ones, dataset 05 depicts the stones in purple and dataset 06 those in cyan.

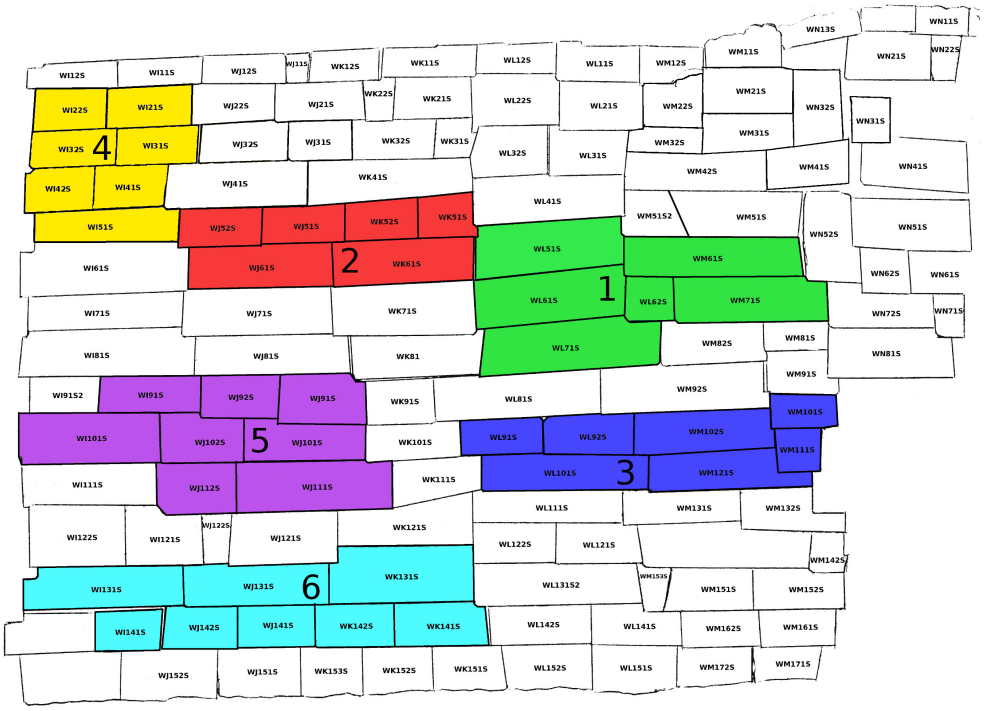
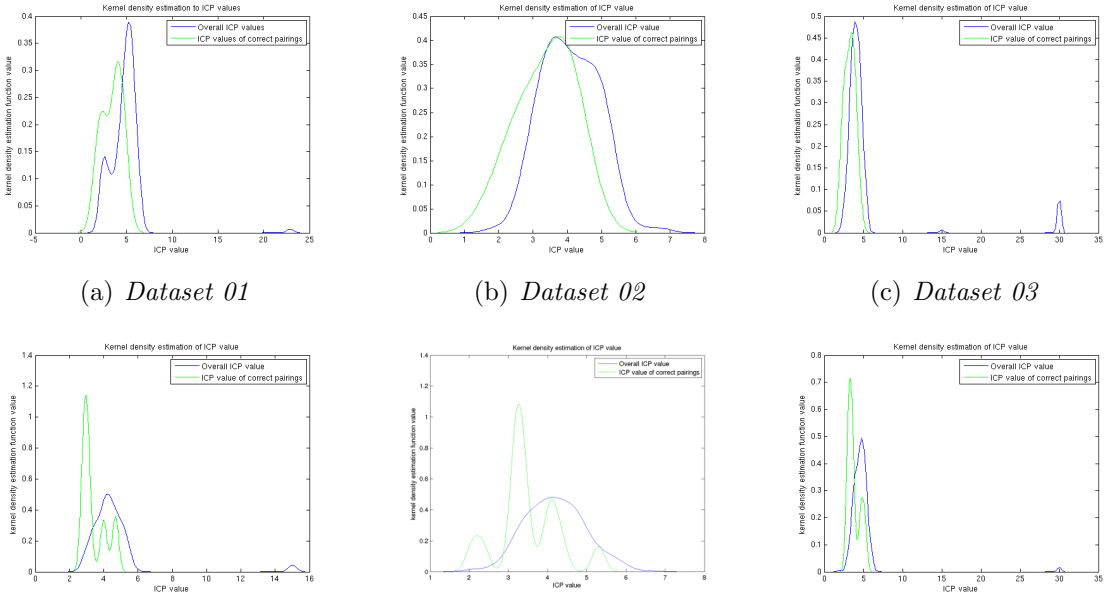


Figure 5.3: *To check the similarity under different conditions six unrelated datasets have been created. The datasets will be referred to with numbers from 1 - 6, where 1 depicts the green one, 2 the red one, 3 the blue one, 4 the yellow one, 5 the purple one and 6 the cyan one.*

[33]Some of the stones were heavily weathered or feature a very dark surface. The acquisition equipment used for scanning of the stones is sensitive to darker parts leading to possible holes in the resulting 3D data.

Figure 5.4 shows how the ICP performs in the six test cases. The blue curve depicts the ICP value of all possible pairings in the dataset plotted against their kernel density estimation. The green line illustrates this for correct pairings only. In all cases correct pairings usually have a low ICP value although a clear distinction between correct values and overall values cannot be drawn.



(a) *Dataset 01* (b) *Dataset 02* (c) *Dataset 03*

(d) *Dataset 04* (e) *Dataset 05* (f) *Dataset 06*

Figure 5.4: *Performance of the ICP algorithm tested on different datasets: the probability of finding correct pairings within low ICP values is higher than for wrong pairings yet there is no clear offset between the two. The datasets are shown in figure 5.3.*

5.2 Similarity analysis based on geometric distances

Described in the introduction to this chapter similarity is measured in terms of geometric distances. This section presents functions that are able to make statements about point clouds and their geometric distance. The explanatory power of those functions is checked for the already mentioned six selected datasets, see figure 5.3. In the following it is assumed that two stones are in contact as discussed in chapter 4.6, aligned using the ICP as presented in the previous section and their associated contact point clouds are given as $p_i \in \mathbb{R}^3$, $P = (p_0, p_1, ..., p_N), 0 \leq i \leq N$, and $q_j \in \mathbb{R}^3$, $Q = (q_0, q_1, ..., q_M), 0 \leq j \leq M$. N, M denote the size of the point clouds. The minimum Euclidean distance (5.4) between two points $p \in P$ and $q \in Q$ is defined as:

$$d(p, q) = \min \|p - q\|_2^2 \tag{5.4}$$

The points (p, q) then will be called closest points.

There are several measures which can be applied. From the possible generally known and well accepted approaches the some were excluded from further evaluation. These are distance measures such as the Manhattan distance, the Maximum distance or the Pearson correlation. The reason is that the point cloud extracted from Khmer temple stones did not meet the prerequisites (e.g. the pearson correlation, often used for similarity analysis, can not be applied as the point clouds are not random variables). Other methods such as a heat kernel estimation were excluded as their implementatory costs are too high.

5.2.1 Frechet distance

The Frechet distance is considered as *"a suitable measure for the similarity of shapes represented by parameterized curves or surfaces"*[34], and is also considered for a pairing of Khmer temple stones. This distance is mainly used for polygonal curves and measures their resemblance in arbitrary dimensions. Its counterpart for triangulated surfaces is proven to be NP-hard. Compared to the Hausdorff distance, see 5.2.3, which concentrates on the set of points on both curves, the Frechet distance is able to represent the course of the curves.

The formal definition of the Fréchet distance is as follows, based on [AB10]. Let f, g be parameterizations of a curve or surface

$$f, g : [0, 1]^k \to \mathbb{R}^d, \quad k \in \{1, 2\}, d \geq k \tag{5.5}$$

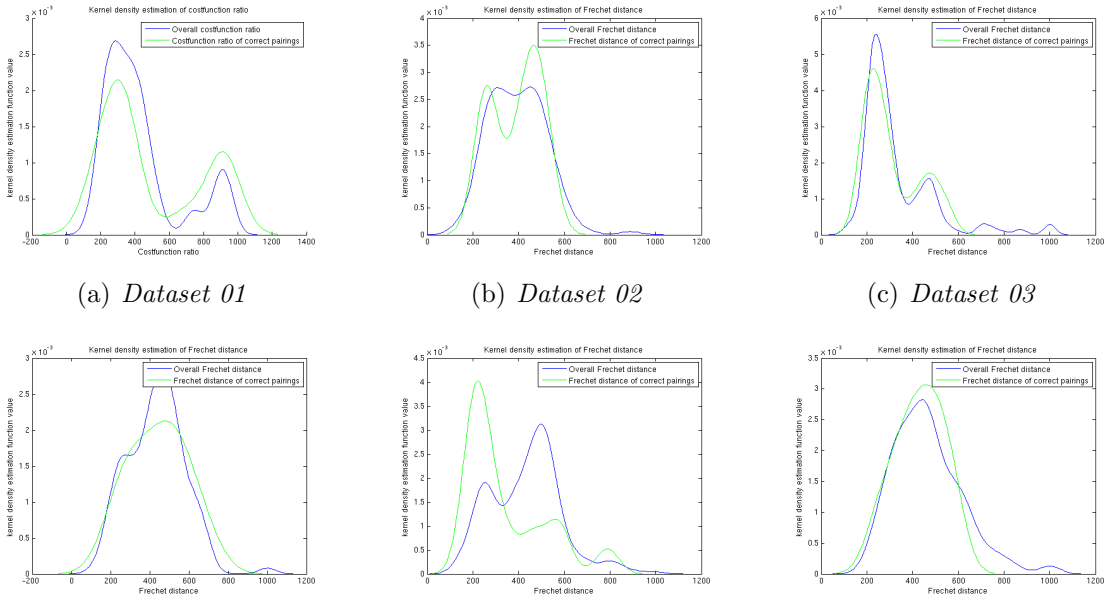
Then the Fréchet distance (5.6) is given as

$$\delta_F(f, g) = \inf_{\sigma:[0,1]^k \to [0,1]^k} \max_{t \in [0,1]^k} \|f(t) - g(\sigma(t))\| \tag{5.6}$$

where σ is the reparametrization ranging over all homeomorphisms preserving the orientation.

Figure 5.5 illustrates the performance of the Frechet distance in the six test cases. The blue curve depicts the Frechet distance of all possible pairings in the dataset plotted against the kernel density estimation. The green line illustrates this for correct pairings only. Except for the case of dataset 05 the distribution of correct pairings closely resembles that of all pairings meaning no clear distinction can be drawn.

[34] H. Alt and M. Buchin, Can we compute the similarity between surfaces?, Journal of discrete and computational geometry (DCG) 43 (2010), p78



(a) *Dataset 01* (b) *Dataset 02* (c) *Dataset 03*

(d) *Dataset 04* (e) *Dataset 05* (f) *Dataset 06*

Figure 5.5: *Performance of Frechet distance tested on different datasets: there is no clear offset between the distribution of Frechet distances of all pairings compared to the distribution of Frechet distances of correct pairings. The datasets are shown in figure 5.3.*

5.2.2 Mean and median distance

Among the most widespread methods to measure deviation is probably the usage of the mean or median of a dataset. Given that $p_i \in \{P|i = 0, ..., N\}$ and $q_i \in \{Q|i = 0, ..., M\}$ be two point clouds containing all points of one plane of stone A and B respectively. Then x is determined as the Euclidean distance between the closest points $p_i \in P$ and $q_j \in Q$ respectively. The median m_d of the ordered set X containing all distances x is then defined as:

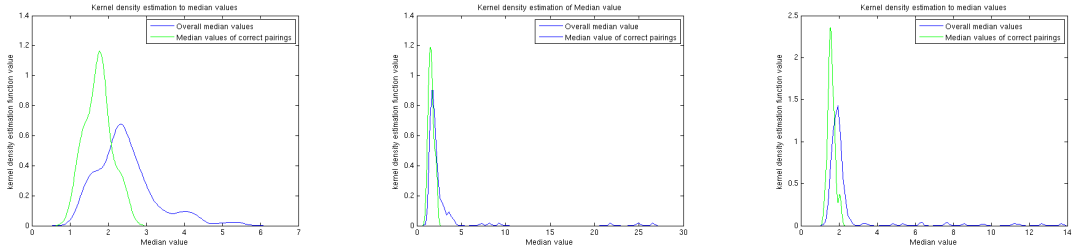
$$m_d = \begin{cases} x_{\frac{n+1}{2}}, & \text{if n uneven} \\ \frac{1}{2}(x_{\frac{n}{2}} + x_{\frac{n}{2}+1}), & \text{if n even} \end{cases} \tag{5.7}$$

The mean m_n of X is defined as follows. The set does not need to be ordered in this case:

$$m_n = \frac{1}{N}\sum_{i}^{N} x_i \tag{5.8}$$

Due to its definition the median is preferred over the mean in the current case as it is more robust to outliers. This criterion determines similarity with regard to the distance of closest points, i.e. the lower the median value of a possible pairing the more likely it is a correct one. Figure 5.6 shows the performance of the median for all six datasets.

The blue curve depicts the median value of all possible pairings in the dataset plotted against the kernel density estimation. The green line illustrates this for correct pairings only. Although there are still outliers it can in general be said that the correct pairings are more likely to have a low median value.



(a) *Dataset 01* (b) *Dataset 02* (c) *Dataset 03*

(d) *Dataset 04* (e) *Dataset 05* (f) *Dataset 06*

Figure 5.6: *Performance of Median tested on different datasets: In general it can be observed that correct pairings are more likely to feature a low median value. The datasets are shown in figure 5.3.*

5.2.3 Hausdorff distance

A distance metric often used in computer vision for the comparison of geometric shapes is the Hausdorff distance. The shapes are represented as sets of points. The Hausdorff distance assigns for every point of one point cloud the distance to its closest point on the other and then takes the maximum of those minima. Its performance is reasonable in practice, yet it can fail if there is too much noise, see [AG99].

The Hausdorff distance is defined as:

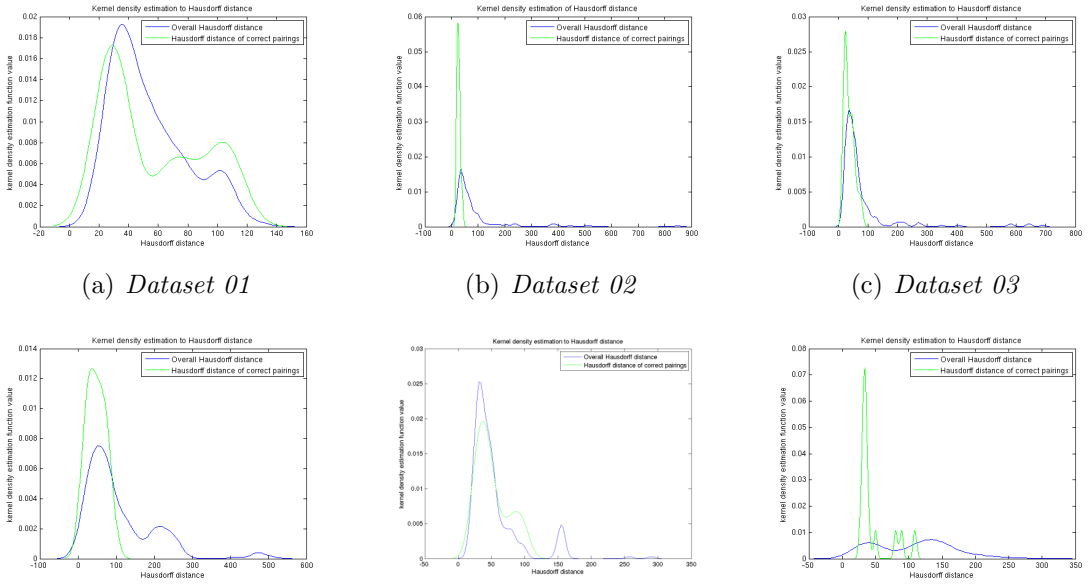
$$h = \max\{\max\{D(p,Q)|p \in P\}, \max\{D(q,P)|q \in Q\}\}, \tag{5.9}$$

with $D(x,K) = \min\{d(x,k)|k \in K\}$.

Variants of the Hausdorff distance are e.g. the average linkage distance or the single linkage distance. The average linkage is the average of all pairwise point-to-point dis-

tances, not necessarily being closest points. Single linkage refers to the maximum of all pairwise point-to-point distances. Both linkage values greatly differ if they compare two large stones and two small stones. Thus their result is not comparable which is why they are skipped from further discussions.

Figure 5.7 shows how the Hausdorff distance performs for the six datasets. As before, the blue curve depicts the value of all possible pairings in the dataset plotted against the kernel density estimation. The green line illustrates this for correct pairings only. It can be observed that for dataset 01 both curves are similar, in case of the second, third and fourth dataset there is a peak where Hausdorff distance values are low with additional large outliers. In case of the last two (dataset 05 and dataset 06) correct pairings clearly stick out as being clustered having a low Hausdorff distance value whereas the overall distribution is more widespread. The differences are due to the different quality of the data.



(a) *Dataset 01* (b) *Dataset 02* (c) *Dataset 03*

(d) *Dataset 04* (e) *Dataset 05* (f) *Dataset 06*

Figure 5.7: *Performance of Hausdorff tested on different datasets: For the datasets 02, 03, 04, 05 and 06 it can be observed that there is a high likeliness that correct pairings feature a short Hausdorff distance. In case of the dataset 01 the distribution of all pairings versus the distribution of correct pairings is similar. The datasets are shown in figure 5.3.*

5.2.4 Distance to Medoid and Centroid

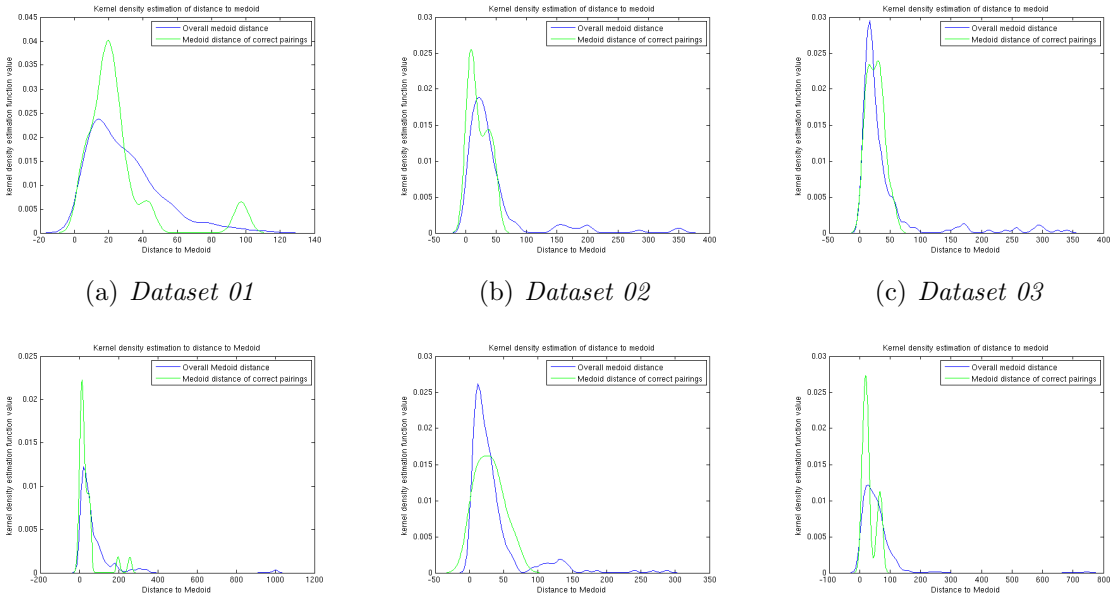
Another measure of similarity is to investigate the scattering of the point clouds from the Khmer temple stones using centroid and medoid. The centroid of a shape is the mean position of all points in this shape, where the medoid of a shape is the median position.

The difference between the two is: A centroid is a mean point but not necessarily part of the shape and the medoid is a median and always an actual point of the point cloud. Given two points $p_i, p_j \in P$ then d_{ij} and their Euclidean distancet the medoid m_{ed} is calculated as

$$m_{ed} = \min_j \frac{1}{N} \sum_i^N d_{ij} \tag{5.10}$$

The similarity measure *distance to medoid* is then computed as the Euclidean distance between the medoids of two point clouds.

Figure 5.8 shows performance of this measure for Khmer temple stones in the six cases. The blue curve depicts the values of all possible pairings in the dataset plotted against the kernel density estimation. The green line illustrates this for correct pairings only. Except for outliers occuring in the number of overall pairings correct pairings and overall pairings cannot clearly be distinguished.



(a) *Dataset 01* (b) *Dataset 02* (c) *Dataset 03*

(d) *Dataset 04* (e) *Dataset 05* (f) *Dataset 06*

Figure 5.8: *Performance of distance to medoid tested on different datasets: The distribution of correct pairings compared to all pairings closely resembles each other in all six datasets. The datasets are shown in figure 5.3.*

5.2.5 2D costfunction

In 2001 Kong and Kimia [KK01] introduced an algorithm that is able to solve a 2D puzzle and at the same time is computationally very efficient. Their proposal was a so-called *costfunction* for the task of finding correct matchings in a dataset of 2D puzzles

pieces. The costfunction extracts and considers the boundary curve of each piece and compares all boundary curves against each other. It is a comparison of the length of a corresponding section and the change in the orientation angle. Using dynamic programming this approach can be applied to compare all segments of all curves with each other and only takes about three seconds. Sticking to the original notation, let $A = (a_1, a_2, ..., a_N)$ and $B = (b_1, b_2, ...b_M)$ be cyclic lists of the two contours respectively. A table consisting of M rows and N columns at most containing the costs of partial matches is build using dynamic programming. The starting point pairs are denoted by $(a_i, b_j), 1 \leq i \leq N, 1 \leq j \leq M$. An entry in this table will be called a *cell*. Matching of merged sequences of points $a(m_{w-1}|m_w)$ and $b(n_{w-1}|n_w)$ is indicated by a link between cells (m_{w-1}, n_{w-1}) and (m_w, n_w), i.e. $a(m_{w-1}|m_w)$ denotes the sequence of points $(a(m_{w-1}, a(m_{w-1}+1), ..., a(m_w))$, the same goes for $b(n_{w-1}|n_w)$. A linked sequence of cells $((m_0, n_0), (m_1, n_1), ..., (m_t, n_t))$ is called a *path* indicating a partial match, where $m_0 = i, n_0 = j, m_0 < m_1 < ... < m_t, n_0 < n_1 < ... < n_t$. In each entry cell (m_w, n_w) the values $cost, u_w, v_w$ are contained. Cost refers to the partially accumulated matching cost up to that entry, the indices of the parent entry are represented by u_w and v_w. The final cost $D(A, B, i, j)$ of matching the two contours A and B starting from a_i and b_j respectively ist then given as:

$$D(A, B, i, j) = \sum_{w=1}^{t} \psi(a(m_{w-1}|m_w), b(n_{w-1}|n_w)), \tag{5.11}$$

where $\psi(a, b) = \eta(a, b) + r\gamma(a, b)$ is the sum of the length of two subcurves of the boundary (η) and the curvature of those subcurves (γ) and thus represents the cost of its arguments. c_w is the ratio of the length of the segments $a(m_{w-1}|m_w)$ and $b(n_{w-1}|n_w)$.

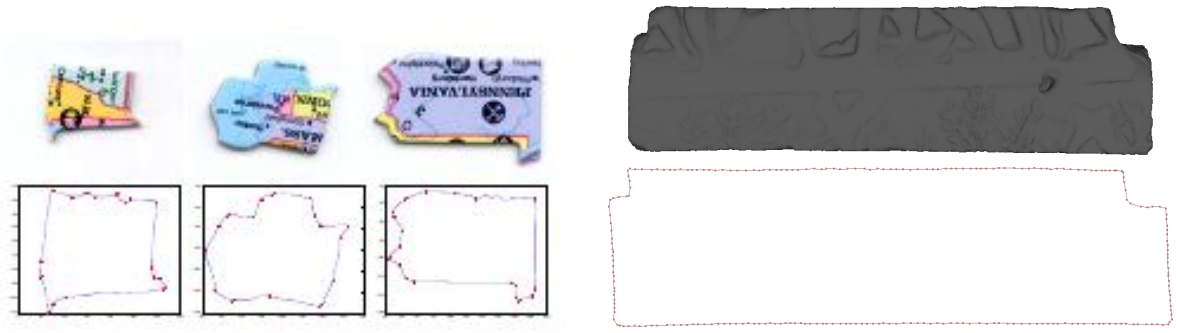
$$\eta(a(m_{w-1}|m_w), b(n_{w-1}|n_w)) = \begin{cases} -2.0/(c_w + 1/c_w), & 0.8 < c_w < 1.2 \\ (c_1 + 1/c_w)/2, & \text{otherwise} \end{cases} \tag{5.12}$$

S_{A_w} and S_{B_w} are the angles of the segments and α_w is the difference of orientation change: $\alpha_w = (S_{A_w} - SA_{w-1}) - (S_{B_w} - S_{B_{w-1}})$.

$$\gamma(a(m_{w-1}|m_w), b(n_{w-1}|n_w)) = \begin{cases} -cos(\alpha_w), & -\pi/6 < \alpha_w < \pi/6 \\ |\alpha_w|, & \text{otherwise} \end{cases} \tag{5.13}$$

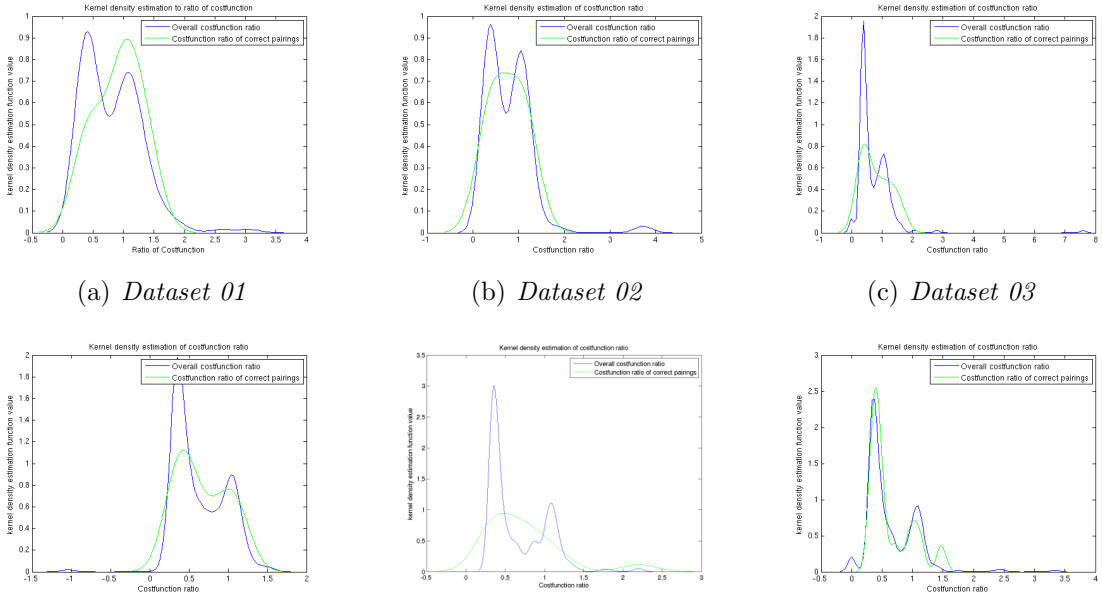
As the sandstone blocks from the Khmer bas-relief wall have very tight vertical and horizontal joints this approach is a possible measure how well two stones fit together. Applying this approach to a 2D projection of the stone models, their boundary curves

are extracted and the cost function computed. It has to be pointed out that due to weathering and broken parts the 2D boundary curve can be falsified. In practice it was determined that the ratio r between D and η, i.e. $r = D/\eta$ has the highest explanatory power. Figure 5.9a shows extracted boundary curves used in the original approach and 5.9b shows exemplarily how a boundary from a Khmer temple stone looks like.



(a) *2D boundary from Kong and Kimia, [KK01].* (b) *2D boundary extracted from a Khmer temple stone.*

Figure 5.9: *2D boundaries which are used for a comparison based on a 2D costfunction proposed by Kong and Kimia, [KK01].*



(a) *Dataset 01* (b) *Dataset 02* (c) *Dataset 03*

(d) *Dataset 04* (e) *Dataset 05* (f) *Dataset 06*

Figure 5.10: *Performance of Costfunction tested on different datasets: correct pairings distribute approximately the same as overall pairings making them difficult to detect. The datasets are shown in figure 5.3.*

Figure 5.10 shows the performance of the costfunction for the six selected datasets. The blue curve depicts the costfunction values of all possible pairings in the dataset plotted

against the kernel density estimation. The green line illustrates this for correct pairings only. It is observable that the distribution of overall pairings and the distribution of correct pairings are nearly indistinguishable. Although it seems this renders the method useless, the next section introduces a similarity measure where the employment of equation (5.11) enhances the result.

5.3 A similarity criterion for Khmer temple stones

From the previous discussion of comparing point clouds in two and three dimensions it emerges that the methods are not distinct enough to be used as stand-alone measures which is caused by the fact that wrong matches can score better than correct ones. This section thus presents a novel approach of computing similarity for pairings of Khmer temple stones.

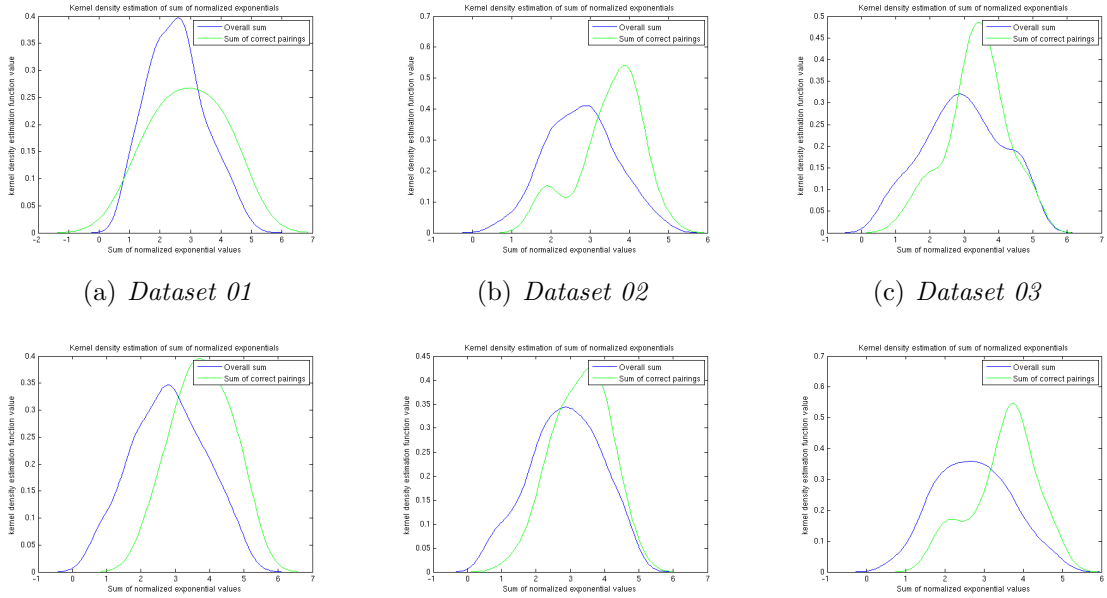
Each of the discussed methods scores on a different scale, which is why first the values were grouped according to the side surfaces they belong to. E.g. if two stones are tested for similarity and each stones has six possible contact faces, there will be 12 sets. Subsequently the values are normalized to be found within a range of $[0, 1]$. This is achieved by applying normalization (5.14) to each of the values. v_i is one of the aforementioned distance measures, $i = 1, ..., k$ is the number of comparisons within each group, e.g. in case there are two stones compared to each other $k = 1$.

$$v_{norm} = \frac{v_i - \min v}{\max v - \min v} \tag{5.14}$$

Finally the normalized values are combined to one similarity value given in (5.15) using an exponential function. The exponential function is used as correct pairings mainly feature comparably low values close to 0 after the normalization which are leveled near 1 after application of the exponential function and therefore become clearly distinguishable.

$$s(v_{norm}) = \sum_{i=1}^{k} \exp(-v_{norm})^2 \tag{5.15}$$

The result of applying equation (5.15) to each of the six datasets is shown in figure 5.11. A significant improvement for the sum of normalized exponential values over the separate values can clearly be observed. Due to the usage of the exponential function, the higher the value, the higher the similarity, i.e. in all six cases the green curve, which depicts the correct pairings, outperforms the blue one, which depicts all possible pairings.



(a) *Dataset 01* (b) *Dataset 02* (c) *Dataset 03*

(d) *Dataset 04* (e) *Dataset 05* (f) *Dataset 06*

Figure 5.11: *Performance of combined similarity values tested on different datasets. The values discussed in the previous subsections are summed up leading to a better identification correct values. The datasets are shown in figure 5.3.*

5.4 Concluding remarks

Summing up the different methods of determining similarity in 2D and 3D shapes or more precisely of determining similarity in case of possibly matching Khmer temple stones they are only able to distinguish a tendency of correct and incorrect pairings. The major contribution of this chapter is if taking the sum of the indiviual measurements the ability to differentiate between a correct and an incorrect match increases considerably. Referring to the quote in chapter 2.4, *"Similarity assessment in 3D cases is usually carried out by generating shape signatures from the 3D models and then comparing these signatures (...). A shape signature could be a graph, a vector or an ordered collection of numeric values. (...)"* the sum can be seen as a shape signature. This shape signature is used in the next step of the puzzle workflow to discriminate correct stone pairings and construct a complete and correct solution of the puzzle. Though a determination of correct pairings is now possible two difficulties remain. Although the sum is a significant advantage over the single methods it is not discriminant enough to clearly determine correct pairings. Chapter 6 introduces an approach that is able to overcome this drawback. The second one is the large number of overall possible pairings shown in table 1. It highlights the overall number of possible pairings versus the number of correct pairings concerning each of the

six datasets. Though correct pairings score comparatitively higher then wrong pairings concentrating on best pairings only still leaves several billion combinatory possibilities. The next chapter will also discuss how this large search space can be reduced and the correct solution be found.

Dataset	Number of overall possible pairings	Number of correct pairings
Green dataset	147	13
Red dataset	135	11
Blue dataset	210	13
Yellow dataset	227	12
Purple dataset	261	15
Cyan dataset	266	14

Table 5.1: The overall number of possible pairings is significantly larger than the number of correct pairings in each dataset.

Chapter 6

Best match determination

“

The possible solutions to a given problem emerge as the leaves of a tree, each node representing a point of deliberation and decision.[35]

”

In the preceding chapters two foundations for automatically and digitally reassembling a Khmer bas-relief temple wall were introduced. First all possibilities to combine two stones were determined and enumerated and second a similarity measure was introduced which is capable of evaluating those pairwise matches. The next step to piece the separate stones together is to solve equation (3.2). Due to the fact that the stones feature a nearly planar surface, which was sometimes difficult to digitally acquire, it is not always assured that a pairing featuring a high similarity value is also a correct pairing. Computing the minimum of (3.2) by selecting only those pairings which score high in the similarity analysis will therefore lead to a false solution. It additionally needs to be ensured that selected pairings form a feasible solution, i.e. the stones are not allowed to intersect themselves, see e.g. figure 6.1.

[35]Niklaus Wirth, Program Development by Stepwise Refinement

Figure 6.1: *A solution with a high similarity score. The red ellipse marks a stone whose final position means to intersect with the other stones in the dataset. While this positioning can be rendered on a computer it is impossible in the real world. Therefore it has to be ensured that such infeasible solutions are sorted out.*

Table 5.1 lists the number of possible pairwise combinations in relation to the number of correct pairings for the six datasets. Due to the high number of potential solutions (e.g. 42 pairwise matches result in more than 42000 different solutions) and the comparatively high similarity values scored by false pairings leading to false solutions, the number of overall pairings should be decreased without decreasing the number of correct pairings. Considering a chosen sample dataset, e.g. dataset 01 which contains six stones (see figure 5.3) every stone is in contact with several of its neighbours. Each connection represents one potential pairwise match of two stones, thus the problem is overdetermined. A complete solution consisting of n stones needs $n - 1$ pairings for a clear positioning of the virtual stones, generally about $2n$ pairings are present. In the following the similarity analysis presented in chapter 5 is enhanced such that it leads to an improved reliability of distinguishing correct pairings and includes a criterion for reducing the overall number of possible matchings. Transforming the results into a graph structure subsequently leads to the optimal solution.

6.1 Similarity based on sum of rankings

The previously discussed options, such as e.g. the Hausdorff distance or the iterative closest point algorithm, showed that geometric criteria are not discriminant enough to be used as a stand-alone measure in order to distinctly find the solution of the Khmer temple reassembly. Thus a combination of the different methods is needed. To this end the similarity function values are grouped into as many sets as there are possible contact faces. Referring to figure 4.14 as an example this results in eight such groups as each of the stones is compared with another stone in two ways.

Chapter 5.3 suggested to combine the similarity values from each of the geometric distance functions and normalize them within the set, see equation (5.14). This approach

achieved a significant improvement over the separate methods. The normalization is required as each distance function scores on a different scale and without its application the values are not comparable. Yet, this standardisation can be falsified if outliers are present. To overcome this drawback and keep the values comparable the function value of each geometric distance method of a pairing in the set is transferred into a rank. Table 6.1 demonstrates this approach for one set of a sample dataset containing eight stones. Each function, in this example the ratio of the cost function, the ICP algorithm, the Median distance and the Hausdorff distance, is ranked according to its functional value, i.e. for example the smallest Median distance is ranked first and the largest Median distance is ranked last. Note that in case of the cost function the largest ratio ranks first. The final decisive similarity value for each of the pairings is computed as the sum of its ranks. In this set, the left side of stone 1 has the highest similarity with the right side of stone 6 with a sum of $2 + 2 + 3 + 4 = 11$.

Pairing	Cost ratio	ICP	Median	Hausdorff	Ranking				Sum
Stone 1/left side – stone 6/right side	2.81	3.401	2.52	27.23	2	2	3	4	11
Stone 1/left side – stone 8/right side	3.07	4.870	5.28	12.69	1	4	6	2	13
Stone 1/left side – stone 2/right side	0.24	6.483	1.57	5.98	7	6	1	1	15
Stone 1/left side – stone 4/right side	1.56	3.221	4.32	30.44	6	1	4	5	16
Stone 1/left side – stone 7/right side	2.62	6.674	3.74	18.66	5	7	2	3	17
Stone 1/left side – stone 5/right side	2.77	5.939	4.33	42.51	3	5	5	6	19
Stone 1/left side – stone 3/right side	2.63	4.453	10.99	45.48	4	3	7	7	21

Table 6.1: Similarity criterion based on the sum of rankings shown for a sample dataset. Each column is ranked according to its function values, e.g. as the Median distance in the third row is the lowest it ranks first. The final similarity value is determined as the sum of the different rankings. In this example the left side of stone 1 has the highest similarity with the right side of stone 6.

A time intensive evaluation was performed for each of the presented geometric distance functions for all six datasets: The placement scoring of correct pairings in each function was analyzed. A combination of ICP, Median, Hausdorff distance and 2D costfunction will yield the best performance and score high rankings for correct pairings and lower rankings for incorrect pairings. Additionally, this allowed to introduce a weighting factor w. Thus the similarity value based rank sum per pairing $\tau(p) = r_s(p)$ (see equation (3.1)) is now given as 6.1, where r is the rank sum per pairing and set, p the pairing, g the set M indicates the Median, I the ICP, H the Hausdorff distance and Cf the costfunction.

$$\tau(p)_g = w_M \cdot r_M(p) + w_I \cdot r_I(p) + w_H \cdot r_H(p) + w_{Cf} \cdot r_{Cf}(p). \tag{6.1}$$

In practice optimal results were achieved setting $w_m = 0.4, w_I = 0.3, w_H = 0.2$ and $w_{cf} = 0.1$, as Median distance and ICP algorithm scored best, followed by Hausdorff distance and ratio of the costfunction.

To validate the performance of this approach figure 6.2 shows the distribution of correct pairings for datasets 1-6 in comparison to the distribution of overall pairings. The matchings are sorted according to their sum of rankings. For all six datasets more than 75% of correct pairings can be found within the first 25% of all sorted possible pairwise combinations. This demonstrates the high quality and robustness of the developed similarity analysis based on the sum of rank ordered geometric distances. Only in case of dataset 05 a fraction of correct pairings is contained inside the last quarter of all matchings. Although dataset 05 has a higher error-proneness due to difficult cirumstances while digitally acquiring the stones (see section 3.3) its overall performance is still reasonable.

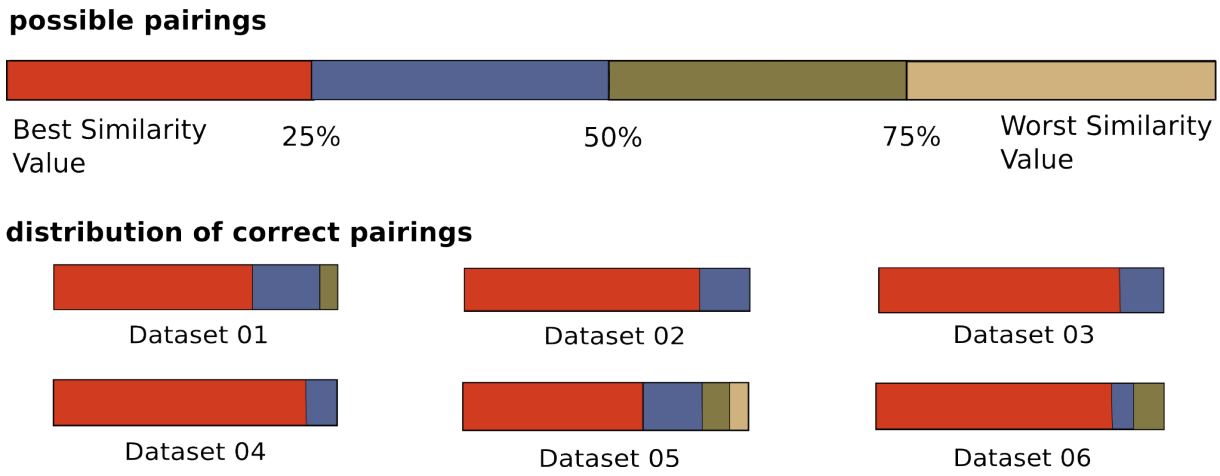


Figure 6.2: *The similarity analysis was applied to six different datasets. The figure demonstrates how the correct pairings distribute within the sorted possible pairings. It can be observed that for every dataset the correct pairings concentrate in the first quarter of all matchings.*

6.2 Pruning

The problem of forming a solution from the possible pairings is overdetermined, thus an advanced pruning technique sorting out the most unlikely pairings is applied to reduce the combinatorial and computational complexity. The rank sum tables from the previous step are the basis in order to achieve this reduction. Those tables are generated for all sides of all stones. The entries of the highest scoring rank sums in each table are considered for corresponding matches. If a corresponding pairing is found, it is stored for further

evaluation, otherwise it will be omitted. Mathematically spoken, for a dataset of z stones there are q ordered sets L as well as q ordered subsets $L_{sub} \subseteq L$. L_{sub} contains the m highest ranking entries of L. The entries of L are computed according to table 6.1 and equation (6.1) and L is given in equation (6.2). Recall that i, j denote the stones and t_i, t_j their respective sides.

$$L_b = \{r_{s_j}(i, t_{i_b}, j, t_j) \mid 1 \leq j \leq z, j \neq i\}, \tag{6.2}$$

with $1 \leq b \leq q$ and $1 \leq i \leq z$. A pairing $p_{ij} = (i, t_i, j, t_j)$ will only be considered for further analysis, if $p_{ij} \in L_{w_{sub}}$, $p_{ji} \in L_{v_{sub}}$ and equation 6.3 holds, i.e.

$$p_{ij} = p_{ji}, \tag{6.3}$$

where $v \neq w, 1 \leq v, w \leq q$. For example, in case of table 6.1 the sets of stone 6/right side, stone 8/right side, stone 2/right side, stone 4/right side and stone 7/right side were compared if the pairing "stone 1/left side" can be found within the first five entries. To create subsets L_{sub}, m is in practice set to 5 for the six test datasets, which is why this method is called *best out of five* in the course of this chapter. If L_{sub} contains less than five entries too many pairings are sorted out, i.e. the pairings not being omitted are not able to form a complete solution, whereas more than five entries does not omit enough pairings to reasonably reduce the problem size.

The presented innovative pruning technique thus reduces the number of possible pairwise combinations to a maximum of five possibilities per side. E.g. in case of dataset 01 the number of possible pairings is reduced from 144 to 42. A statistical analysis was performed to validate this step: More than 80% of possible but wrong pairings were sorted out whereas at the same time only 15% of correct pairings were rejected. Figure 6.3 displays for each of the six datasets the relation of all pairings versus correct pairings before and after this best out of five approach. Generally, in less than a second most wrong pairings are omitted and nearly all correct pairings are kept.

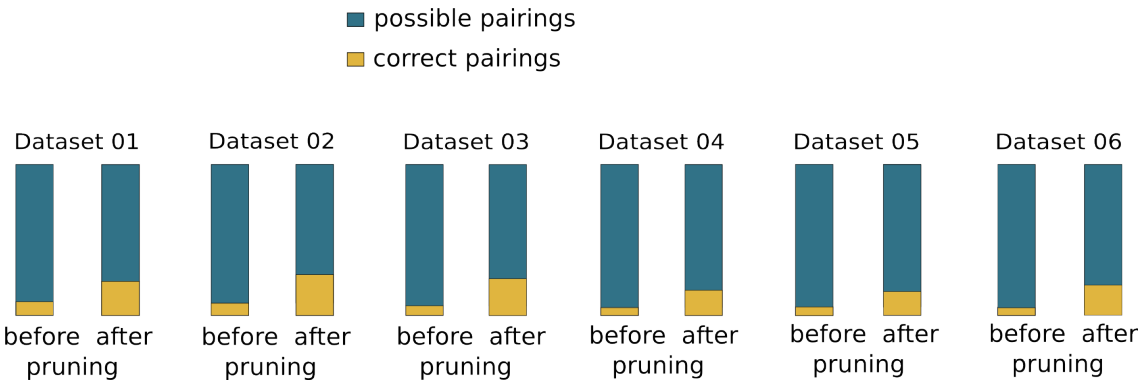


Figure 6.3: *After applying the best out of five approach the number of wrong pairwise matchings reduces by 76.21% on average whereas the number of correct pairings is reduced by only 19% on average. The columns show the ratio of all pairings compaired to correct pairings before and after the pruning step has been applied.*

6.3 Graph-based approaches

The enhanced similarity measure based on the sum of rankings and the pruning which is applied subsequently reduces the high number of possible pairings but the correct solution needs yet to be found. Although many wrong pairings are sorted out it is still possible that the solution which minimizes the rank sum of all pairwise matches, is not the correct one. Finding this correct solution for Khmer temple stones thus equals to finding the optimal combination of pairings. This combinatorial problem can be solved by applying graph-based approaches. The stones are considered as the vertices of the graph and the pairwise combinations as its edges. The following definitions are closely related to Diestel [Die06] and Jungnickel [Jun13].

Definition: Graph, weighted graph

A *graph* is a pair $G = (V, E)$ of disjoint sets with $E \subseteq [V]^2$. The elements of V are called the *vertices* or *nodes* of G and the elements of E are its *edges*. A graph can be visually realized by drawing the vertices as points and the edges as connecting lines between the points. Two vertices x and y of G are called *adjacent* if $xy \in E(G)$. Two edges are called adjacent if they share a common vertex. If a cost $c : E \to \mathbb{R}$ is assigned to the edges, the graph will be called a *weighted graph*.

Definition: acyclic, connected

A sequence $(e_1, ..., e_i)$ of edges in a graph G with vertices $v_i \in G$ such that $e_i = v_{i-1}v_i, i = 1, ..., n$ is called a *walk*. A walk for wich $v_0 = v_n$, the e_i and v_j are distinct except for $v_0 = v_n$, and $n \geq 3$ holds, is a *cycle*. Two vertices v_k, v_p of a graph G are *connected*, if there exists a walk starting at v_k and ending at v_p. A graph is acyclic, if it does not contain any cycles.

Definition: bipartite graph

If a partition $V = A \cup B, A \cap B = \emptyset$ of the edgeset $V(G)$ exists, such that each edge $e \in E(G)$ has exactly one vertex in A and one vertex in B, then this graph is *bipartite*.

Definition: distance, shortest path

The *distance* $d(v, w)$ between two vertices $v, w \in G$ is defined as:

$$d(v,w) = \begin{cases} \infty & \text{if } m \text{ is not accessible} \\ \min(w(P))|\ P \text{ is a path from } w \text{ to } v & \text{otherwise} \end{cases} \tag{6.4}$$

Any path achieving the minimum of equation 6.4 is a *shortest path*.

Definition: Tree, spanning tree, minimum spanning tree

Let T be a graph with n vertices. If any two of the following three conditions hold, the third is implied:

(i) T is connected.

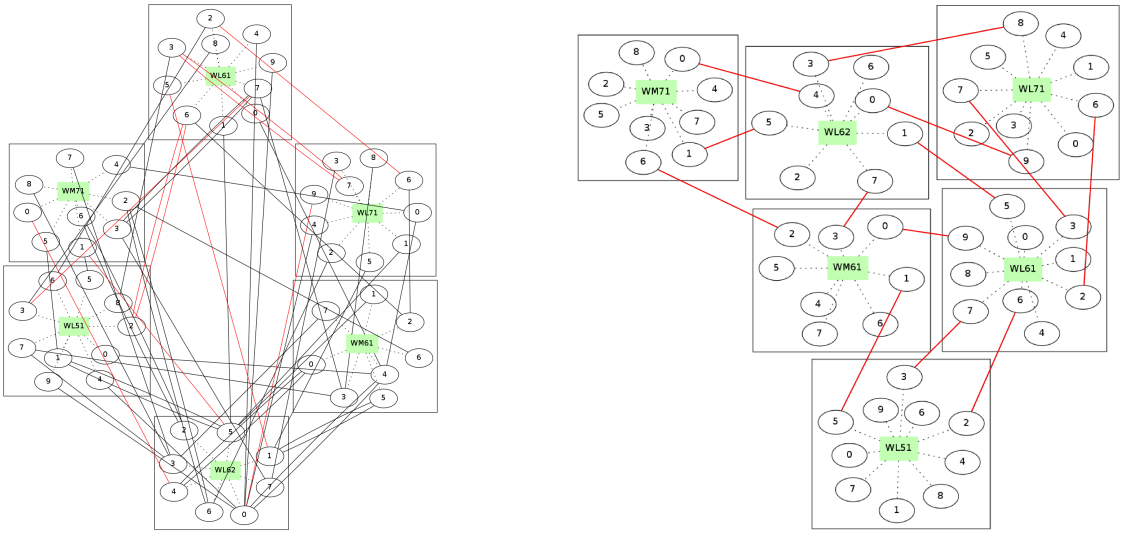
(ii) T is acyclic.

(iii) T has $n - 1$ edges.

Then T will be called a *tree*. For a proof see [Jun13]. A *spanning tree* is graph S if it is a spanning subgraph of a graph G and is a tree. If G is a weighted graph and S minimizes

the sum of all including edges, then S is a *minimum spanning tree*.

To apply graph theory algorithms to the reassembling of fallen Khmer temple stones the given datasets need to be transferred into a graph. As already suggested, the stones are the nodes of the graph and the connections are the edges of the graph. Several intricacies of this approach have to be considered. Firstly, starting with all possible pairwise matchings and transferring them into a graph, it is possible that several edges between the same two nodes (i.e. stones) occur. This appears, if after the application of the pruning step more than one possibility to match these stones remains. Those edges are therefore difficult to distinguish. Secondly, regarding possible pairings as the graphs vertices, the information of which side belongs to which stone is lost and the result is a set containing only pairs of adjacent vertices. Thirdly, by assigning each stone a set of numbers (i.e. stone 1 = 0, stone 2 = 10, stone 3 = 20, side 1 (any stone) = 0, side 2 (any stone) = 1, ...) and the numbers are added up, the resulting combinations are unique. E.g. number 11 is assigned to side 2 of stone 1. Fourthly, to overcome the drawback of the first two suggestions combinations of them can be used. The stones as well as their sides are represented as nodes in the resulting graph, edges between sides of different stones (denoting possible pairings) have a weight larger than zero and edges between the stone and its sides (denoting to which stone a side belongs to) have zero weight. This last approach is illustrated for dataset 01 in figure 6.4, where (a) shows all possible pairings in black and enhances the correct ones in red and (b) shows the correct ones.



(a) *all pairings* (b) *correct pairings*

Figure 6.4: *For dataset 01 figure 6.4(a) shows all possible pairings in black and enhances the correct ones in red, and figure 6.4(b) shows the correct pairings.*

After the transformation of possible pairwise matches into a graph according to the fourth approach the task to find the correct solution remains. In order to solve this task the similarity values of each pairing are used. They are applied as weights to the edges of a graph. Edges with minimal weights are preferred for calculation under the assumption that the correct solution minimizes the combined sum of ranks of possible matchings. Additionally, only a minimal set of pairings is needed which is able to connect all stones. This equals to the graph theoretical problem of finding a minimum spanning tree. There exist two greedy algorithms that are able to find the minimum spanning tree of a graph, one variant is called *Prim's minimum spanning tree*, the other one *Kruskal's minimum spanning tree*. The problem is $NP - hard$ thus the algorithms cannot be proven to be correct. Both have a complexity of $O(n^2)$.

A minimum spanning tree using Prim's algorithm has been applied to the six datasets. In all cases, the solution suggested by the minimum spanning tree algorithm contains correct and incorrect pairings. Due to the fact that it is possible for incorrect pairings to score high similarity values the solution suggested by Prim's algorithm is not the correct one, although it already contains correct pairings. Thus, a minimum spanning tree approach is not able to determine the correct solution in the current case. Figure 6.5 exemplarily shows the result of applying Prim's minimum spanning tree algorithm to the first dataset.

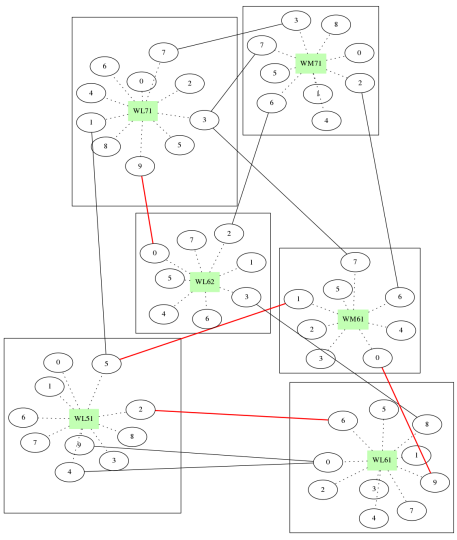


Figure 6.5: *Applying a minimum spanning tree algorithm to pairwise possible matchings with the aim of constructing a complete solution. The edges between a stone and its sides have weight zero and the edges between two sides, being the possible pairwise matchings have their similarity rank value assigned as weight. The red edges mark correct pairings. It is observable, that a mixture of correct and incorrect pairings forms the minimum spanning tree.*

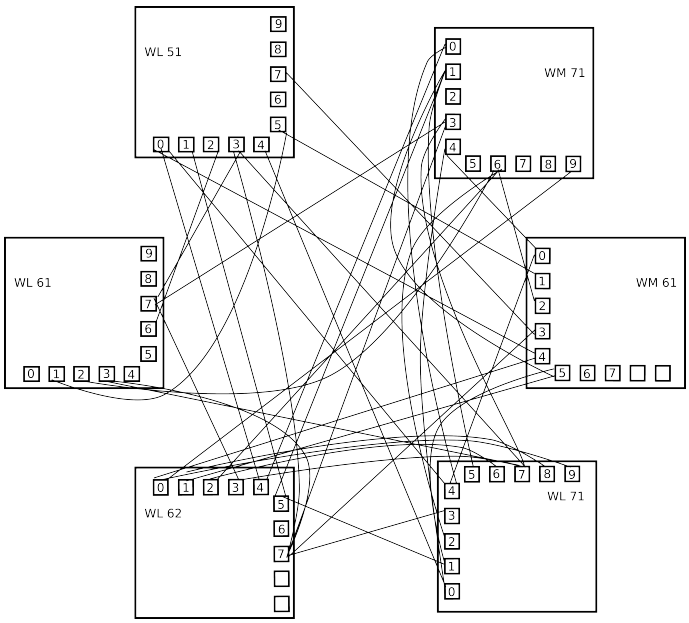


Figure 6.6: *Examining the applicability of k-partite graphs to the Khmer temple reassembly. As k-partite graph theory examines ways to find completely connected graphs and the solution to the 3D puzzle is to find a minimal connected graph,this approach fails.*

As weight minimization algorithms such as the minimum spanning tree are not applicable the suitability of k-partite graphs was examined. A k-partite graph is a generalization of a bi-partite graph where the graph can be partitioned into k separate sets of edges. Each stone and its sides form one of the k sets, see figure 6.6. Yet, in k-partite graph theory the search focus is laid on looking for completely connected graphs as e.g. the Turan graph whereas we are looking for the minimal complete correct solution, which is a subset of all remaining possible pairings.

6.4 Solution Construction

The puzzle workflow up to this point (MVB computation, matching enumeration, similarity analysis, find minimum rank sum) was applied to the synthetic test data presented in section 3.4. In this ideal case the correct solution is also the one minimizing problem (3.2) thus solving the reassembly task. In case of the real-world Khmer temple stones, however, the preceding sections taught that false pairings can score higher than correct ones which is why classical approaches are inapplicable. Subsequently, a new concept was developed, that is capable of non-ambiguously determining the correct solution. It consists of two steps, first all possible and feasible solutions are enumerated using a tree structure, the so-called *solution tree* and second a force-directed graph layout is applied. As not only

the numerically minimal solution is considered but to also the final arrangement of the stones is taken into consideration the correct solution featuring the optimal layout and a low accumulated rank sum is uniquely detected. The following subsections elaborate how this method works, present advantages and drawbacks and show results.

6.4.1 Solution tree

To be able to obtain all possible solutions from the pairwise matchings a tree structure is applied. A vertex (also called node) of the tree is a possible pairing $p(i, t_i, j, t_j)$ of two stones. An edge of the tree is a connection between two pairings (= vertices) iff one stone occurs in both vertices, e.g. $v_1 = p(i, t_i, j, t_j)$ and $v_2 = p(i, t_i, k, t_k)$. The root node of the tree can be chosen to be any one of the potential matches. Each level of the tree represents all possibilities how one pairing can be added to the already existing partial solution. Thus the adding procedure is terminated if either all stones of the dataset are in use or no further pairings are available. The depth of the tree is determined by the number of stones in the dataset and, assuming there are enough pairings, is given as: $n - 1$, where n is the number of stones in the dataset. Before another vertex is added to a partial solution an intersection test is performed. It determines whether the new stone possibly intersects with the already given stones (see e.g. figure 6.1). If this is the case then this branch is cut as it represents an impossible solution. Thus the deepest level of the solution tree contains only the remaing feasible and complete solutions. An excerpt of the resulting tree is shown in figure 6.7. As the final result of the solution tree still provides a large number of possible solutions (e.g. 42 possible pairings in case of dataset 01 lead to about 1313 possible solutions) those solutions are sorted according to their accumulated rank sum, i.e. the sum of each pairings' rank. In practice it was determined that the correct solution does not feature the lowest but still has a comparatitevely low accumulated rank sum. Table 6.2 shows the minimal accumulated rank sum of the six evaluated datasets as well the correct and the maximal accumulated rank sums. To reduce the large number of possible solutions, the minimal and maximal accumulated rank sum is computed and the median determined. All solutions with a rank sum above the median are subsequently filtered out. This resulting dataset size is also given in table 6.2. The remaining solutions are then evaluated by a force-directed graph layout which is discussed in the next section.

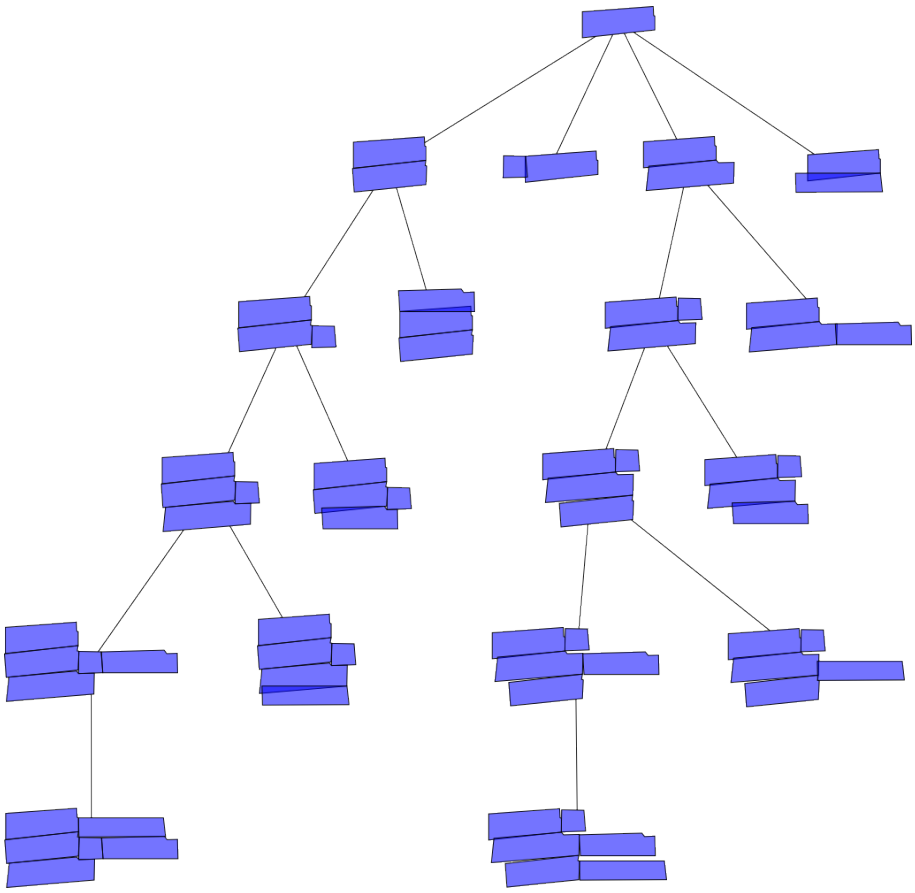


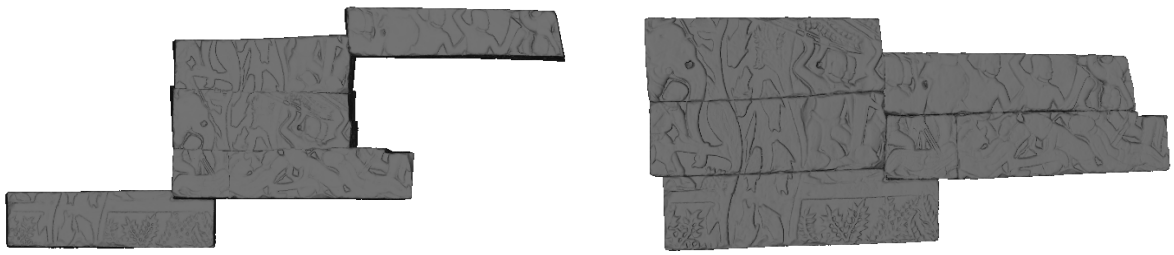
Figure 6.7: *An excerpt of the solution tree for dataset 01. Potential pairwise matches of two Khmer temples stones are the nodes of this tree. In each level another pairing is added to the already existing partial solution if it passes an intersection test. This is necessary to avoid impossible solutions, see figure 6.1 for an example. If the intersection test is failed the branch in question is cut. In the deepest level of the tree, only feasible solutions are present. The depth of the tree is determined by the number of stones in the dataset.*

	Dataset 01	Dataset 02	Dataset 03	Dataset 04	Dataset 05	Dataset 06
Minimal Rank sum	12	10	15	12	17	17
Correct Rank sum	14	11	17	16	30	20
Maximal Rank sum	41	43	54	52	60	54
Remaining solutions	1313	2530	7602	11489	13385	32472

Table 6.2: Results of the solution tree for the six datasets. It shows the minimal sum of accumulated ranks, the rank sum of the correct solution as well as the maximum sum of ranks. Additionally the number of remaining solutions after filtering out solutions whose accumulated rank sum is above the median is given.

Figure 6.8.a shows the solution with the minimal accumulated rank sum for dataset 01 and figure 6.8.b shows the correct reassembly of the same dataset. From an algorithmical point of view it is difficult to tell which of the two is the correct and optimal

reassembly as in both cases the stones do not intersect and the arrangement seems valid. The minimal rank sum criterion therefore serves as a presorting condition, whose output is taken as input for the force-directed graph drawing algorithm in the last step of the puzzle reassembly. The application of the solution tree as a presorting step is necessary to ensure impossible solutions are removed as well as possible solutions kept for further consideration.



(a) (b)

Figure 6.8: *Both figures show a possible solution for dataset 01. (a) depicts the minimal accumulated rank sum solution and (b) the correct one.*

6.4.2 Force-directed graph drawing

Although a graph can be drawn using points and lines a proper visual embedding is not always straightforward. The importance lies in the fact that the graphs needs to be correct as well as easily understandable. To this end the research field of graph drawing *"adresses the problem of constructing geometric representations of graphs, networks, and related combinatorial structures."*[36] The search for the best solution within the many that remained after the construction of the solution tree is supported and simplified by making use of so-called force-directed graph drawing (FDGD) algorithms.

Usually graphs are used to illustrate the relationship between objects. Although there are no strict rules applying to graph drawing it is in general agreed upon an even distribution of vertices, minimal edge crossing and symmetry. The introduction and application of force-directed methods in graph drawing relates to Tuttes work in 1963 [Tut63] and to Fruchterman and Reingold [FR91]. Tutte showed that a polyhedral graph can be drawn in a plane by assigning attractive forces like e.g. springs to the edges and then let the system settle to an equilibrium state. Fruchtermann and Reingold introduced an algorithm that produces *"aesthetically-pleasing, two dimensional pictures of graphs by doing simplified simulations of physical systems."*[37] FDGD methods are therefore flexible, easy to implement and result in pleasant drawings. Their basic idea is to model the graph as a

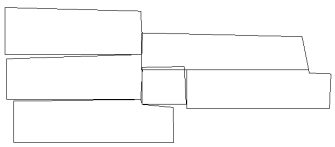
[36](Graph Drawing: algorithms for the visualization of graphs, Battista, 1999, p.vii)
[37]Fruchtermann and Reingold, p. 1

system of bodies with interacting forces where the aim is to find the optimal arrangement minimizing the energy of the system. This is achieved by modeling the forces as springs attracting and repelling each other. Typically, attracting springs are modelled based on Hooke's law and repelling springs are based on Coulomb's law. After having modelled the nodes, edges and springs the whole system can be simulated like a physical system. The springs then either pull the nodes closer together or push them further apart until an equilibrium is reached. The final positions of this system are then used to actually draw the graph. Methods for finding an equilibrium state can e.g. be stress majorization, energy minimization or simulated annealing. The difficulty of FDGD approaches is that the objective function usually has many local minima especially for large graphs.

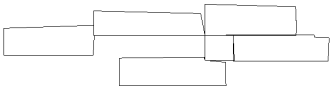
Applying a FDGD method to find the best solution is done by keeping the stones as vertices of the graph and adding only those pairings as edges (or springs) which are occuring in the complete solution. Taking the results from the solution tree, where complete solutions with an accumulated rank sum above the median are already sorted out, the FDGD approach is able to reliably determine the correct solution within all remaining ones by assigning a scoring value to each solution. The score is calculated using the accumulated rank sum (derived from applying the solution tree), the overlapping area and the gap distance between the stones. Table 6.3 shows the performance of the force directed graph drawing method for the correct solution of the six datasets in relation to the solution having the minimal rank sum. Clearly, the score of the correct solution outperforms the score of the solution featuring the lowest rank sum. It has to be pointed out, that the starting positions for each stone are chosen at random yet in all cases in the final position the algorithm assigns a clearly distinguishable score to the correct solution. Figure 6.9(a),(c),(e),(g),(i),(k) shows the resulting graph for the correct solution of the six datasets and figure 6.9(b),(d),(f),(h),(j),(l) shows the resulting graph for the minimal rank sum solution.

	Dataset 01	Dataset 02	Dataset 03	Dataset 04	Dataset 05	Dataset 06
Minimal Rank sum solution	21588	19641	14941	20510	85425	42502
Correct solution	15670	15616	14271	17526	43990	25644

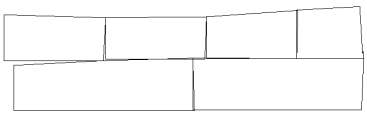
Table 6.3: Results of the force directed graph drawing for the six datasets. It shows the score of the force-directed graph drawing approach for the solution having minimal rank sum and the correct solution.



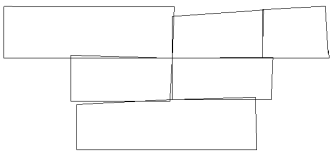
(a) *correct solution, dataset 01*



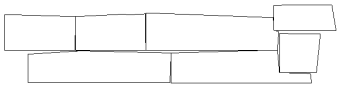
(b) *minimal rank sum solution, dataset 01*



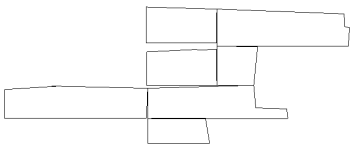
(c) *correct solution, dataset 02*



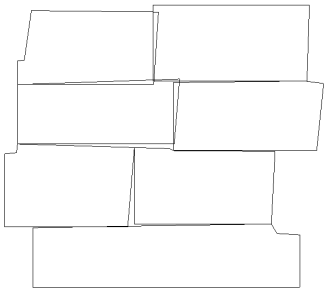
(d) *minimal rank sum solution, dataset 02*



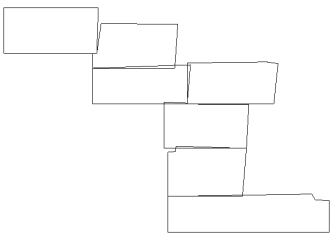
(e) *correct solution, dataset 03*



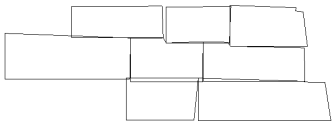
(f) *minimal rank sum solution, dataset 03*



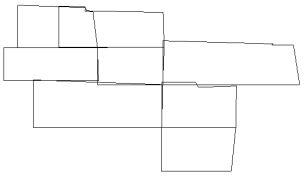
(g) *correct solution, dataset 04*



(h) *minimal rank sum solution, dataset 04*

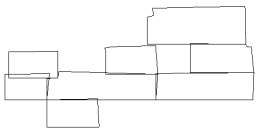


(i) *correct solution, dataset 05*



(j) *minimal rank sum solution, dataset 05*

(k) *correct solution, dataset 06*



(l) *minimal rank sum solution, dataset 06*

Figure 6.9: *The result of the FDGD approach for the six datasets in case of the correct solution (a),(c),(e),(g),(i),(k) and the minimal rank sum solution (b),(d),(f),(h),(j),(l).*

Chapter 7

Results

> *When I'm working on a problem, I never think about beauty. I think only how to solve the problem. But when I have finished, if the solution is not beautiful, I know it is wrong.*[38]

The performance and quality of the methods which have been developed and employed during the course of this thesis have already been discussed in chapters 4, 5 and 6 accordingly. The concept of a minimal volume box was explicitly designed and developed for Khmer temple stones in order to have an accurate and storage efficient description of each stone. From figure 4.9 it emerges that although an MVB is an underestimate of the original volume it is more accurate than an oriented bounding box. Exploiting this easy to handle structure all plausible pairwise combinations can be enumerated and stored prior to the similarity analysis. The determination of highly similar pairings and the discrimination of dissimilar combinations is conducted using a novel approach based on combining ranks of different geometric distance functions explained in section 5 and 6.1. Based on their ordered accumulated rank sum a pruning removes the most improbable matchings and retains those with a high similarity. The definite non-ambiguous identification of the correct solution is enabled by constructing complete solutions from all remaining pairwise matchings and applying a force-directed graph drawing algorithm.

Through this approach two major contributions have been achieved. First, it is now possible to virtually analyze digitized Khmer temple stones in order to reassemble them as a wall and second the correct positioning can be exactly identified.

[38] Richard Buckminster Fuller

Each step of the presented reassembly algorithm was tested on six different datasets containing six to eight stones of different measurement quality and the results were presented in the according chapters. All computations have been performed on an Intel i7 3.07Ghz CPU with 24 Gb RAM. Whenever possible, a parallelization was applied using OpenMP. The software framework the algorithm is integrated into is called *scifer*.

(a) *Dataset 01*

(b) *Dataset 02*

(c) *Dataset 03*

(d) *Dataset 04*

(e) *Dataset 05*

(f) *Dataset 06*

Figure 7.1: *The six reassembled test datasets are illustrated. The stones are correctly reassembled upon the application of the puzzle workflow.*

Scifer is developed and used by the Computer Graphics and Visualization workgroup at the Interdisciplinary Center for Scientific Computing at the Heidelberg University.

Figure 7.1 shows the virtually reassembled stones of the six test datasets after the last step of the puzzling workflow. A sketch of the datasets and their location in the original wall is shown in figure 5.3. Figure 7.1.a shows the reassembly of dataset 01, figure 7.1.b dataset 02, figure 7.1.c dataset 03, figure 7.1.d dataset 04, figure 7.1.e dataset 05 and figure 7.1.f dataset 06.

The remaining of this chapter discusses the complete puzzle pipeline for three larger datasets, see figure 7.2. Dataset 07 contains 16 stones marked in light green, dataset 08 contains 25 stones which consist of the ones from dataset 07 plus the stones with darker green color and dataset 09 which contains all marked stones. The aim is to examine the behaviour of the puzzling workflow in case of larger datasets.

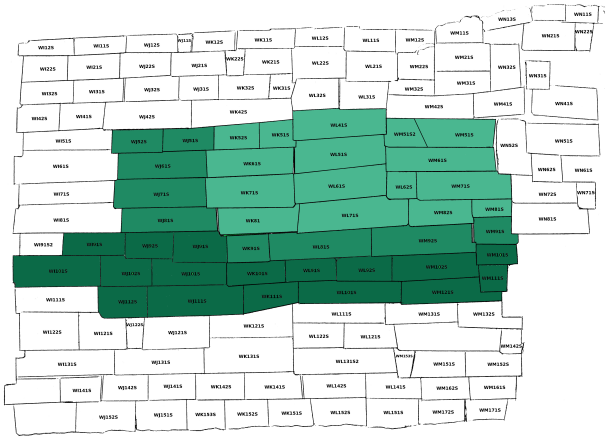


Figure 7.2: *Testing the puzzling workflow on larger datasets: Dataset 07 contains 16 stones marked in light green, dataset 08 contains 25 stones which consist of the stones with darker green color in addition to the ones from dataset 07 and dataset 09 which contains all stones marked in green.*

7.1 Minimal Volume Boxes and Possibility Enumeration

For each of the digitized Khmer temple stones its minimal volume box is computed according to chapter 4. This step can be precomputed for all stones at once and the result can be stored for further evaluation as the minimal volume box of a stone remains undisturbed as long as the underlying stone is unchanged. As the edges of the stone feature

a significantly higher curvature than the flat side surfaces this step can be performed on a low resolution version of the digital stones which speeds up the calculation. The computational amount of this step is $O(n)$. Enumerating all possible pairwise combinations to perform a similarity analysis results in 1076 pairings for dataset 07, 3090 pairwise matchings for dataset 08 and 8162 pairwise combinations for dataset 09.

7.2 Similarity analysis

Computation of the similarity analysis requires $O(n^2)$ time. For dataset 07 64.1% of correct pairings can be found within the first quarter of all pairings and 1.28% in the last quarter, for dataset 08 65.83% of correct pairings are in the first quarter and 9.17% in the last quarter. In case of dataset 09 there are 72.86% of correct pairings which can be found within the first 25% of overall pairings and 4.76% correct pairings within the last quarter. In all cases on average more than two thirds of correct pairings can be found within the first quarter of all possible pairwise matchings and only an average of 5.07% score low in the rank sum based similarity analysis. In conclusion it can be deducted that the rank sum based similarity analysis is very robust even for larger datasets.

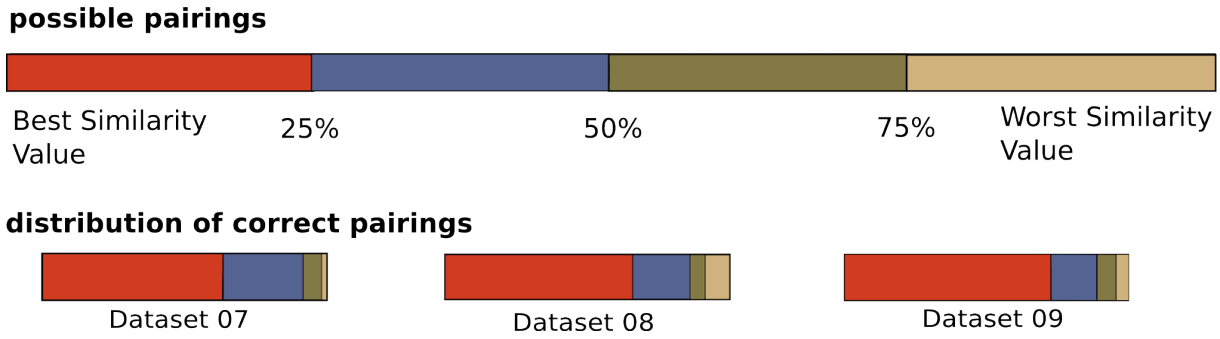


Figure 7.3: *The distribution of correct pairings within all possible pairings of datasets 07, 08, 09.*

7.3 Best out of five and solution tree

Constructing the solution tree leads in general to a high number of possible complete solutions. The tree is based on a list of remaining possible pairwise matchings after the performance of the pruning step determining the five highest ranking pairings. As described in chapter 6 only a minimal set of correct pairings is needed for a correct solution, thus the problem of reassembling the stones is overdetermined. In reverse this means that even if some correct pairings are omitted by the best out of five approach the correct

solution can still be determined. Each stone that is added to the dataset increases the number of pairings as well as the number of wrong pairings with a high rank sum. This increase has to be considered and the pruning step needs to be adjusted to the number n of stones in the dataset in order to fit. In practice the size of the subset L_{sub} (see equation (6.2)) used for the cutting should be chosen to be $\frac{n}{2}$. Table 7.1 depicts the overall correct matches before and after the pruning step has been applied in relation to the number of correct pairings before and after the pruning.

The computational cost of enumerating all solutions using the solution tree depends on the size of the reamining pairings, i.e. the more pairings remaing the more complete solutions can be constructed. If one or several pairings are known in advance, impossible pairings can be deleted from the list prior to the construction of the solution tree. Additionally, the running time of building up the solution tree depends on the number of stones in the dataset. It requires more time if the tree has a larger depth as for every layer of the tree the remaining pairwise possibilities have to be checked. Applying an intersection test, as suggested in chapter 6.4.1 leads to a downsizing of possible solutions thereby speeding up the running time.

Number of	Dataset 07	Dataset 08	Dataset 09
overall matches before pruning	1076	3090	8162
correct matches before pruning	78	120	210
overall matches after pruning	118	248	457
correct matches after pruning	21	38	70

Table 7.1: How many overall matches are reduced due to the pruning step versus the number of correct pairings reduced by the pruning.

7.4 Reassembling the wall using force-directed graph drawing

Applying the force-directed graph drawing that is described in chapter 6 to all possible and plausible complete solutions finally leads to the correct solution. Table 7.2 shows that the score of the force-directed graph drawing approach is significantly higher in case of the correct solution than it is in case of the minimal rank sum solution. The running time depends on the number of stones as a complete solution consisting of n stones contains $n-1$ pairwise matchings. In the force-directed graph drawing these pairwise matchings are taken as the springs and adding one additional stone to the dataset results in one additional spring as well as one additional node in the force-directed graph drawing.

Figure 7.4 shows the final virtually reassembled results for the three datasets 07, 08 and 09.

Concluding, the results in this chapter showed that the puzzle workflow developed in the course of this thesis is able to correctly virtually reassemble stones from a fallen Khmer temple wall whose original position is unknown.

	Dataset 07	Dataset 08	Dataset 09
Minimal Rank sum solution	207737	506846	1365478
Correct solution	104344	457231	733682

Table 7.2: Results of the force directed graph drawing for dataset 07, 08 and 09. It shows the score of the force-directed graph drawing approach for the solution having minimal rank sum and the correct solution.

(a) *dataset 07 with 16 stones*

(b) *dataset 08 with 25 stones*

(c) *dataset 09 with 42 stones*

Figure 7.4: *The reassembled datasets.*

Chapter 8

Conclusion and Future Work

> *Success is the sum of details.*[39]

This thesis presented a novel approach to virtually reassemble a wall of the fallen Khmer temple site Banteay Chhmar, which is a very complex and challenging task. Based on digitally acquired temples stones using a 3D scanner, these virtual stone models support the manual reconstruction done on site. The generally applied approach conducted in-situ uses time-consuming manual labour to suggest potential placements for each stone and verify the assumptions using man-power and cranes. For every error that occurs the process has to be restarted from scratch causing unwanted and unneccessary further deterioration of the stones and additionally risks the safety of the workes who have to carry around the heavy stones. Due to the high complexity this manual approach can take several months.

Making use of the new methods presented in this thesis the manual work can be reduced to an absolute minimum. The stones now only have to be moved twice: once for the digitization and once for the final reassembly. The developed and presented workflow is able to distinctly determine the correct solution. In order to achieve this specific features of Khmer temple stones, namely to be of cuboidal form and the occasional presence of indentations, were exploited. The user interaction is now only needed for the preparation of the stones. Furthermore, the thesis not only provided a puzzle framework but developed a new type of storage saving structure for virtual Khmer temple stones, the minimal

[39] Harvey S. Firestone

volume box (MVB). A MVB closely resembles a virtual stone and can be extended to types of stones used for different parts of the temple.

The MVBs were subsequently used to fast enumerate all pairwise possibilities and a sophisticated similarity measure, based on different geometric distance functions, determined the togetherness of such potential matches. The particular strength of this similarity analysis is the ability to remove pairings featuring a low similarity. By enumerating only pairings with high similarity values through a tree structure all possible and plausible complete solutions were found. Force-directed graph drawing algorithms applied to those complete solutions distinctly found the correct position for each stone in order to rebuild the temple wall.

In conclusion this large-scale 3D puzzle was uniquely solved by applying the presented workflow and it will help to reduce the movement of stones on site as well as reduce the labour-intensity of the work.

8.1 Future work

In the future, the similarity analysis of the presented workflow can be strengthened to also include bas-reliefs on a stone, if present. This has so far not been taken into account as not every stone has a bas-relief applied onto. Some stones did not have one by intention and on some it has been stolen. Evaluating the bas-relief on the stones requires methods from the area of image processing to identify parts of an image spreading over several stones as well as methods from numerical geometry to account for the fact that the bas-relief is not only a 2D image applied on top of the stone but was carved into the stone making it threedimensional.

In case of very large datasets, containing several hundreds of stones, the stones can be presorted according to their width. It was observed for the present part of a temple wall that the two top most and the two bottom most rows of stones are significantly wider than the stones in the central part. Using this feature the stones can be clustered into smaller groups, which reduces the computational complexity. Furthermore advanced data structures and parallelization techniques or the usage of modern-day gpu's[40] can be applied to speed up the computation.

It is possible to extend the developed MVB to other types of stones and the whole workflow can be extended to reassemble other and larger parts of the temple. On top

[40]Grapics processing unit, a highly specialized electronic unit whose architecture is designed to rapidly compute large and massively parallel operations.

of that Khmer temples feature a lot of similarities i.e. the outcome of this thesis can be applied to other temples in former Siam as well. Furthermore, the approach is not limited to temples in South-East Asia but adaptable to fallen Roman, Greek or Mayan temples.

Appendix A

In this appendix all possible pairings how two virtual Khmer temple stones can be combined are shown. Virtual stones which are used as a ground-truth in this thesis and whose orientation in $\mathbb{R}^3$ is known belong to one of the following four classes: having no indentation, having one indentation on the upper right side, having one indentation on the upper left side and having two indentations one on the upper right as well as one on the upper left side.

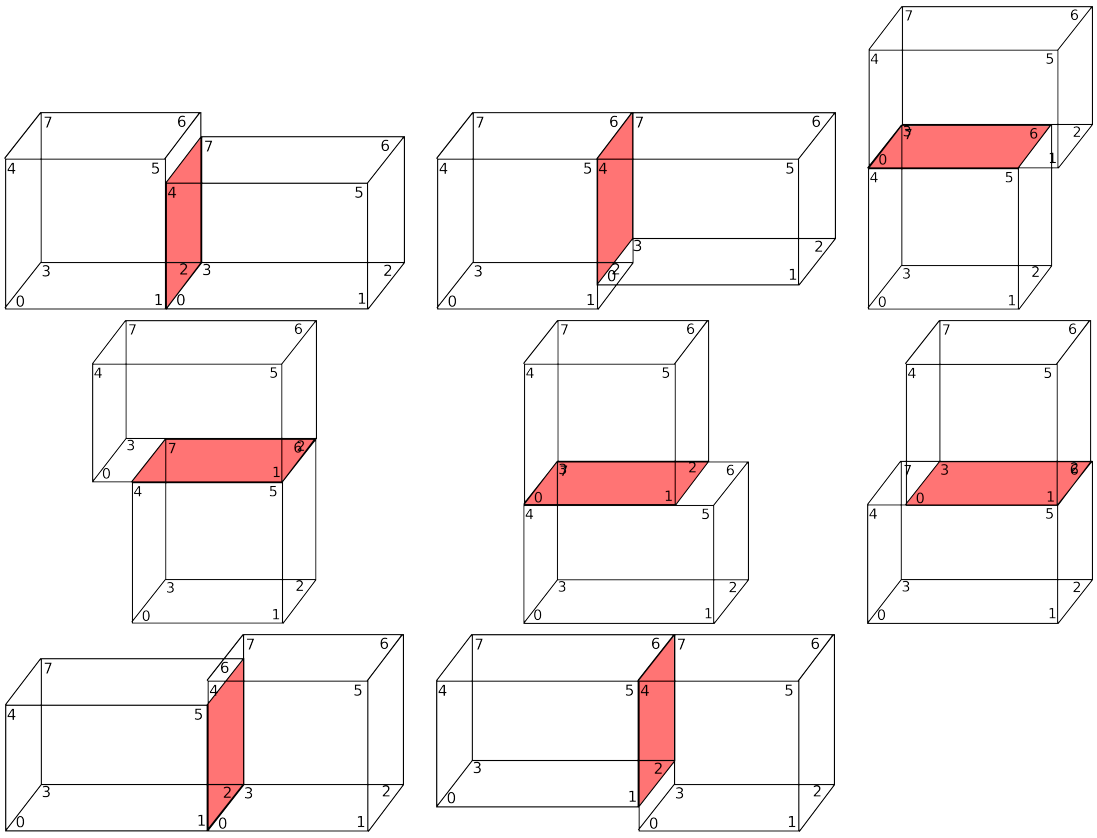


Figure A.1: *Two stones with no indentation are compared to each other.*

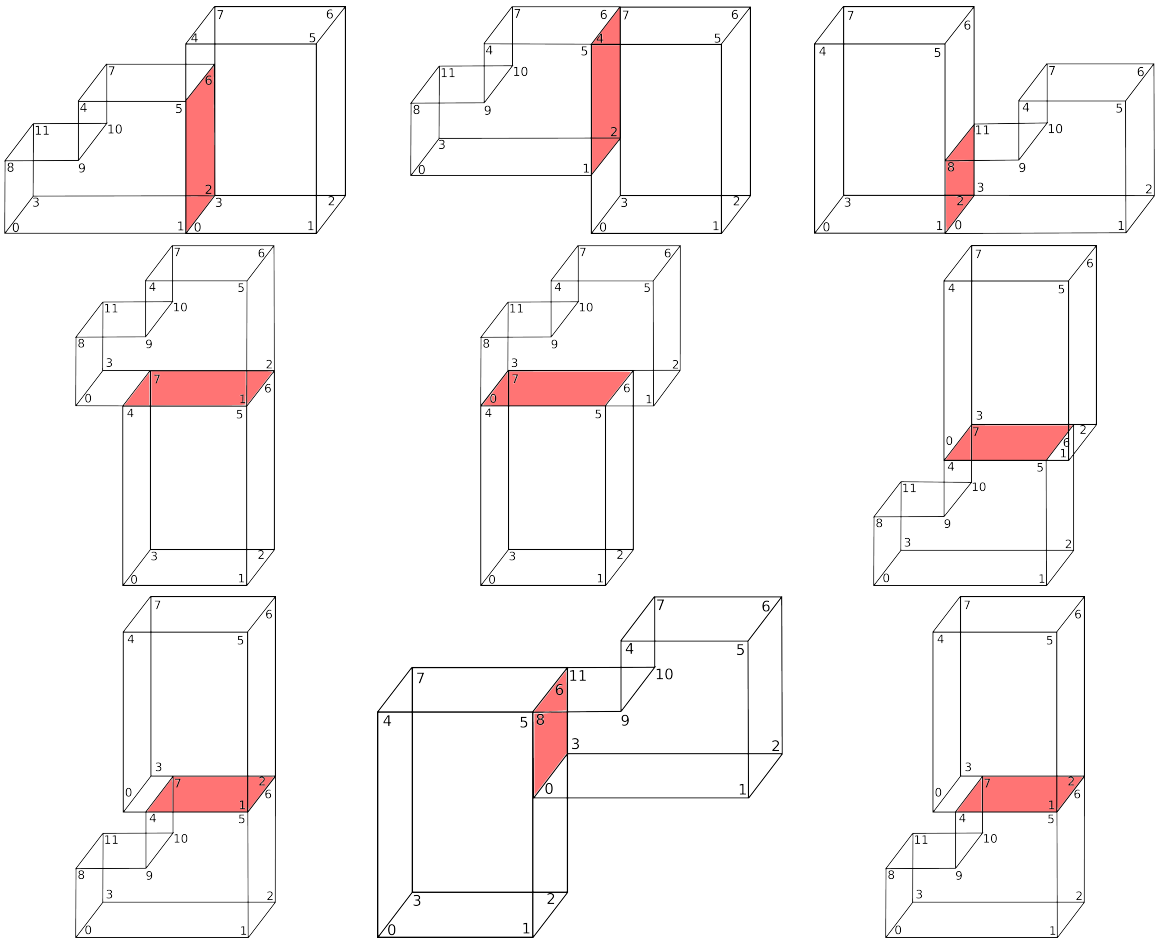


Figure A.2: *A stone with a left indentation is compared to a stone with no indentations.*

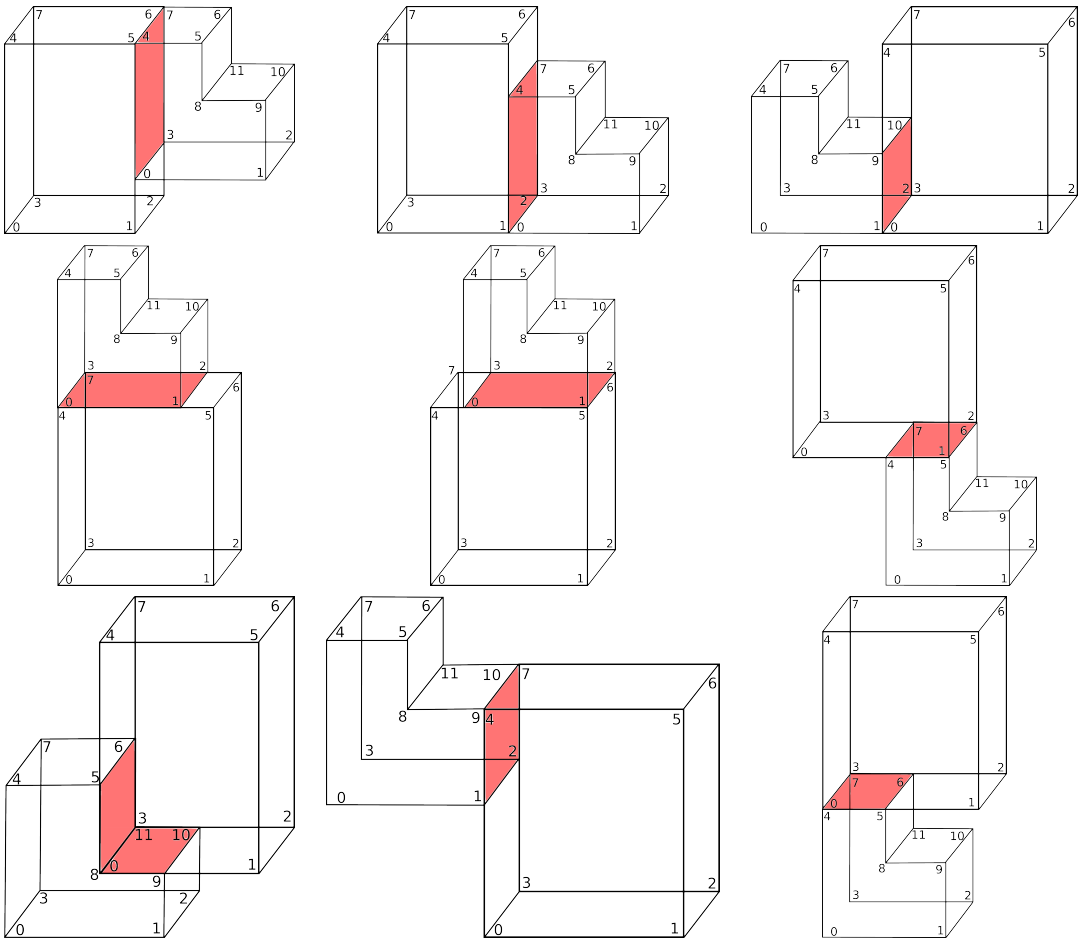


Figure A.3: *A stone with a right indentation is compared to a stone with no indentations.*

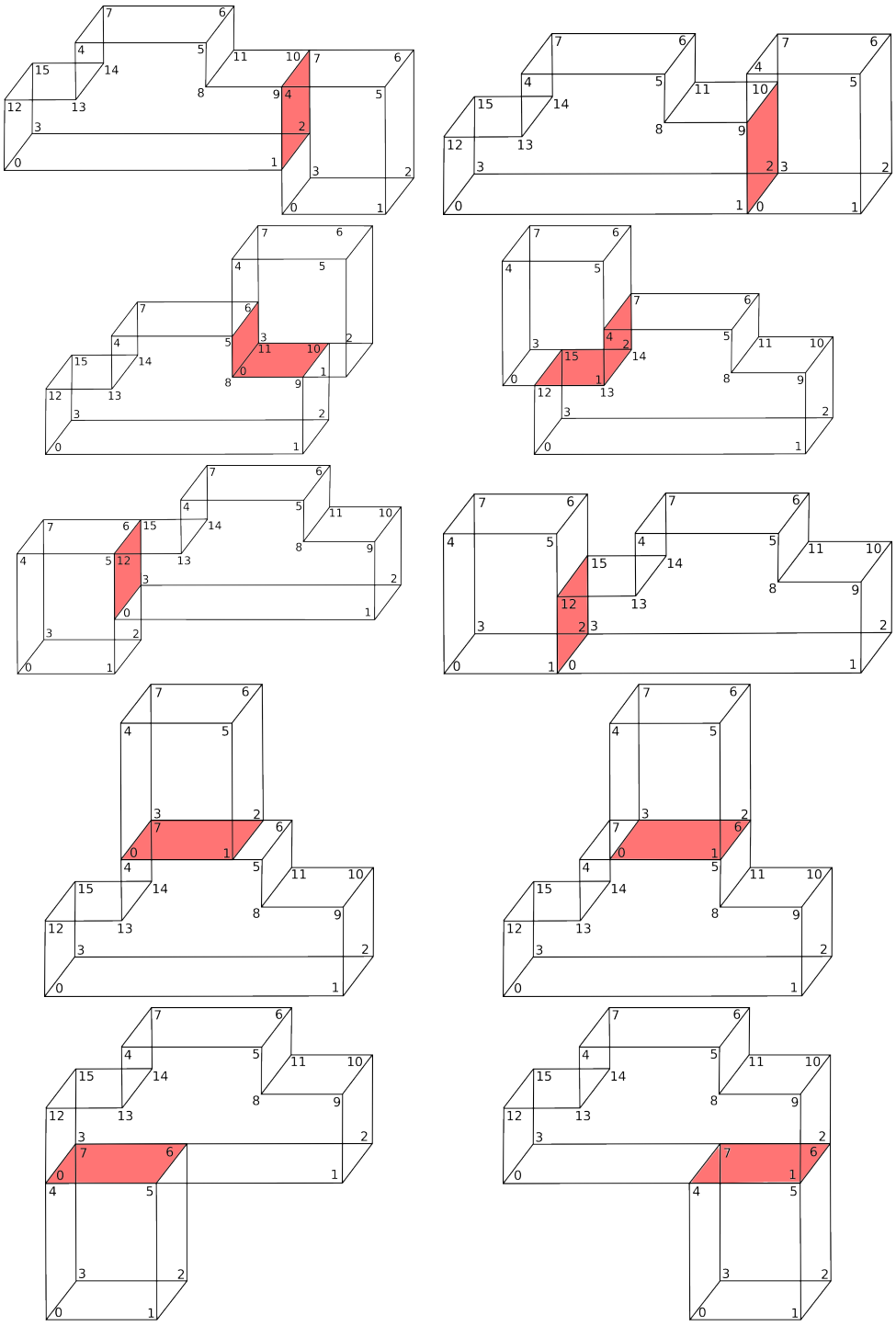


Figure A.4: *A stone with a two indentations is compared to a stone with no indentations.*

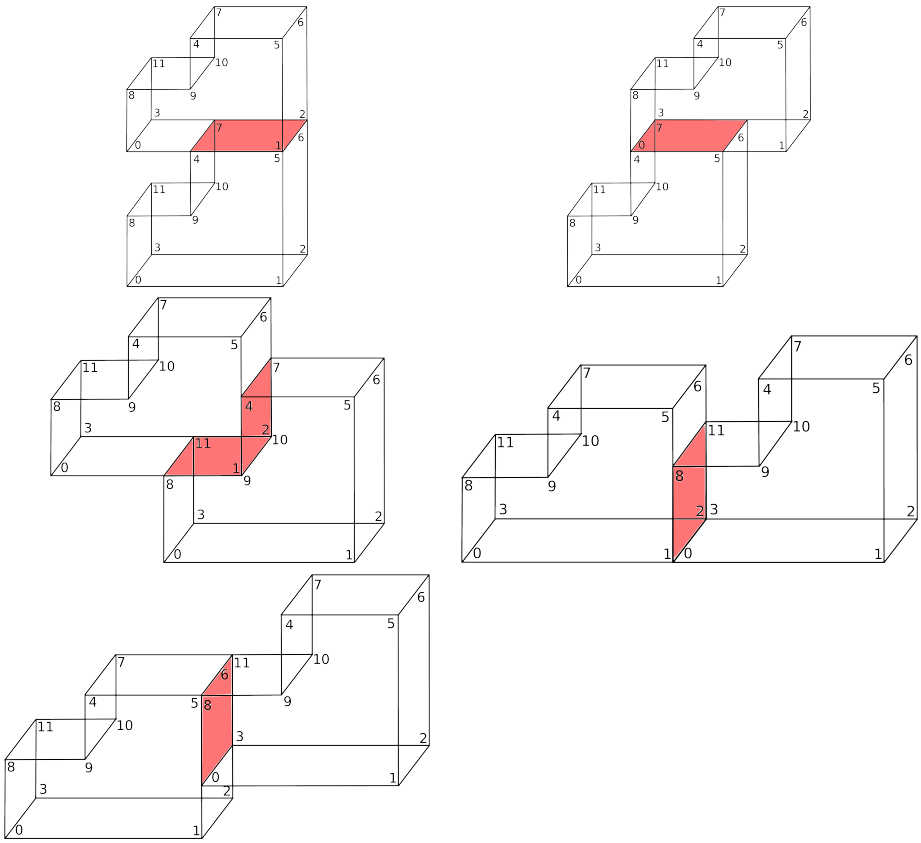


Figure A.5: *Two stones with a left indentation are compared*

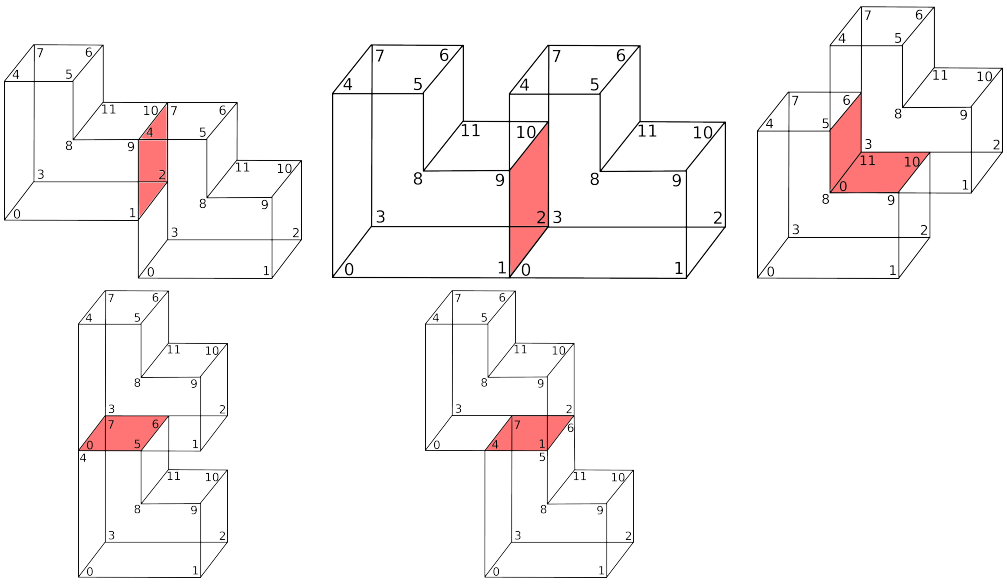


Figure A.6: *Two stones with a right indentation are compared*

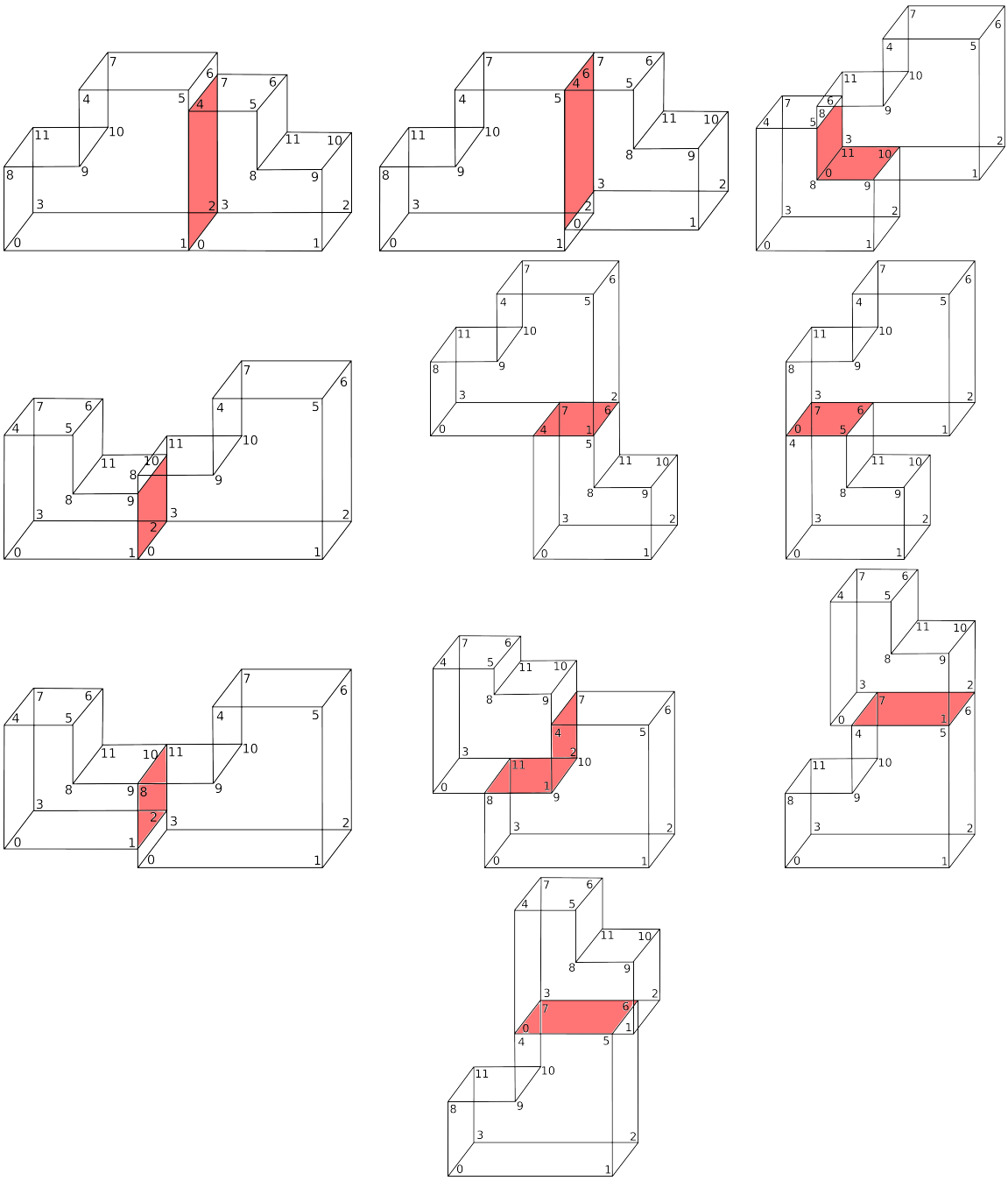


Figure A.7: *A stone with a right indentation is compared to a stone with a left indentation.*

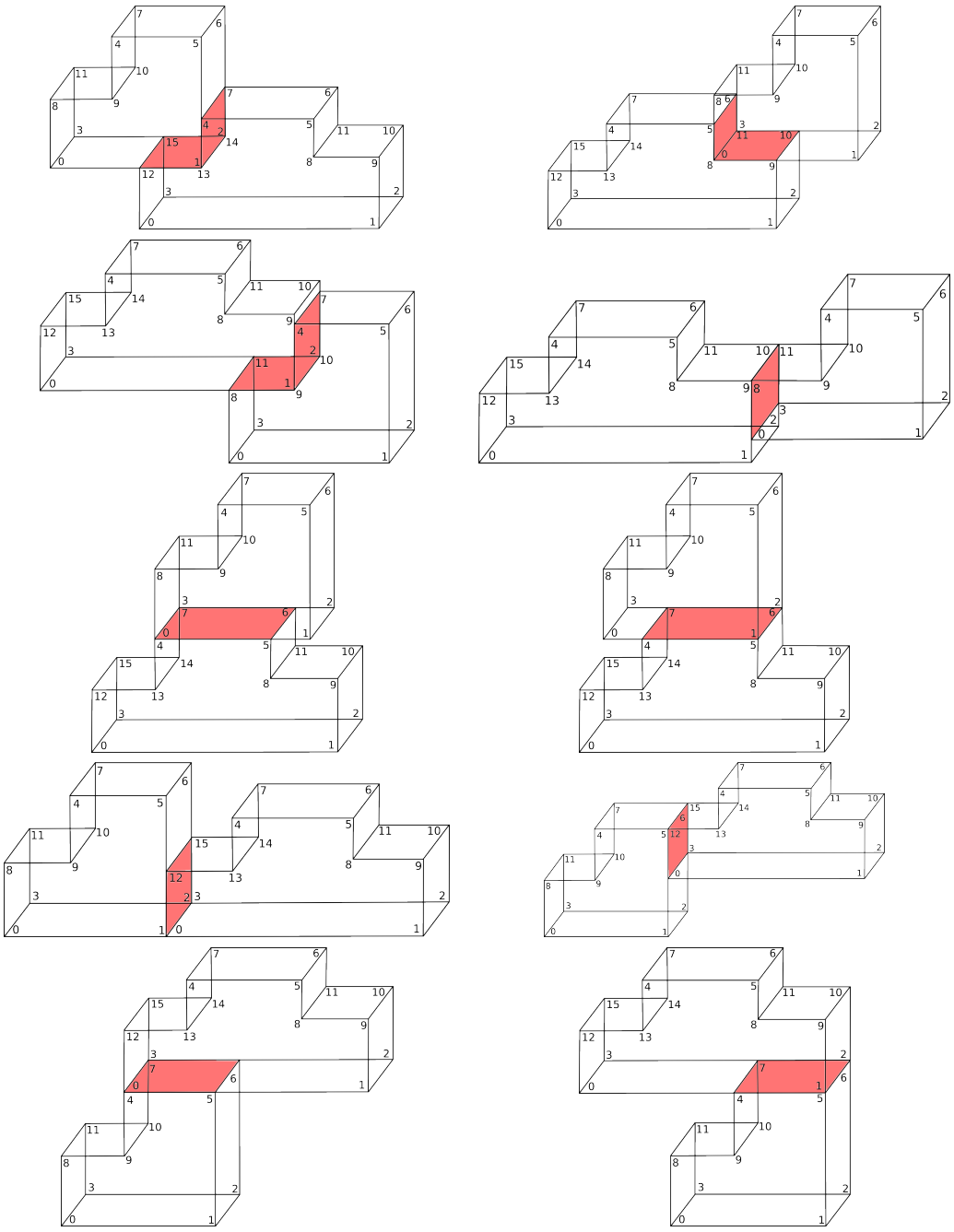


Figure A.8: *A stone with a left indentation is compared to a stone with two indentations.*

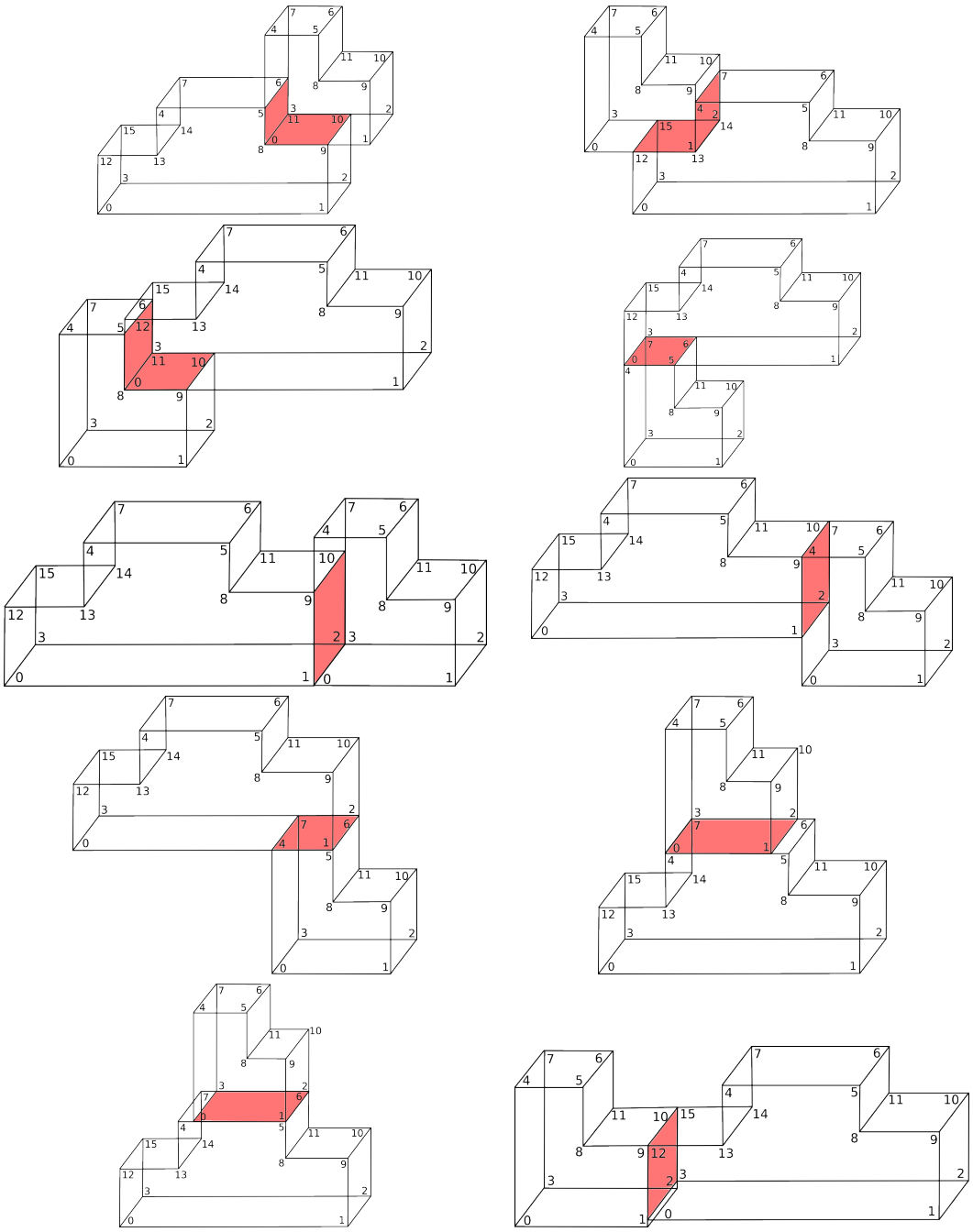


Figure A.9: *A stone with a right indentation is compared to a stone with two indentations.*

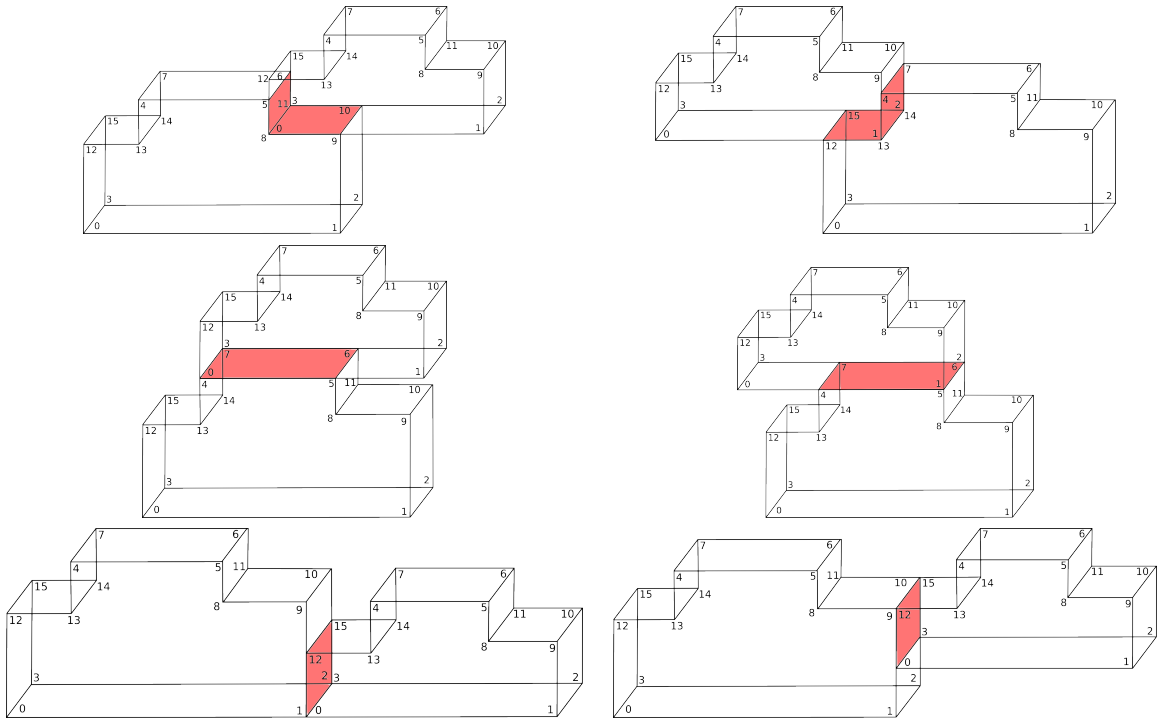


Figure A.10: *Two stones with upper indentations are compared.*

Bibliography

[AA04] A. Adan and M. Adan, *A flexible similarity measure for 3d shape recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004), no. 11, 1507–1520.

[AB10] H. Alt and M. Buchin, *Can we compute the similarity between surfaces?*, Journal of discrete and computational geometry (DCG) **43** (2010), 78–99.

[ABCO+03] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, *Computing and rendering point set surfaces*, IEEE Transactions on Visualization and Computer Graphics **9** (2003), no. 1, 3–15.

[ACKT04] M.O. Altan, T.M. Celikoyan, G. Kemper, and G. Toz, *Balloon photogrammetry for cultural heritage*, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **35** (2004), no. B5, 964–968.

[AG99] H. Alt and L. J. Guibas, *Discrete geometric shapes: Matching, interpolation, and approximation*, ch. 3, pp. 121–153, Elsevier Science, 1999.

[AGI+08] A. Agapiou, A. Georgopoulos, Ch. Ioannidis, N. Doulamis, and M. Ioannides, *Three dimensional reconstruction for cultural heritage visualization. application to the byzantine churches of cyprus*, Proceedings of CAA Conference on the Road to reconstructing the past (2008).

[ASC+13] L. Arbace, E. Sonnino, M. Callieri, M. Dellepiane, M. Fabbri, A. Iaccarino Idelson, and R. Scopigno, *Innovative uses of 3d digital technologies to assist the restoration of a fragmented terracotta statue*, Journal of Cultural Heritage **14** (2013), no. 4, 332–345.

[Aym04] E. Aymonier, *Le cambodge. les provinces siamoises*, vol. III, Paris, 1904.

[Bar01] G. Barequet, *Efficiently approximating the minimum-volume bounding box of a point set in three dimensions*, Journal of Algorithms **38** (2001), no. 1, 91–109.

[BB06] M. Bender and M. Brill, *Computergrafik*, 2nd ed., Hanser, 2006.

[BC87] W.W. R. Ball and H.S.M Coxeter (eds.), *Mathematical recreations and essays*, 13th ed., ch. Chapter 5: Polyhedra, pp. 130–161, New York: Dover, 1987.

[BKS06] D. Bartz, J. T. Klosowski, and D. Staneker, *Tighter bounding volumes for better occlusion culling performance*, Tech. report, University of Tuebingen, Germany IBM, T.J. Watson Research Center, Hawthorne, NY, USA, 2006.

[BLD+12] B. Brown, L. Laken, P. Dutre, L. v. Gool, S. Rusinkiewicz, and T. Weyrich, *Tools for virtual reassembly of fresco fragments*, International Journal of Heritage in the Digital Era **1** (2012), no. 2, 313–330.

[BM92] P.J. Besl and N.D. McKay, *A method for registration of 3d-shapes*, IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992), no. 2, 239–258.

[BMV01] P. Benko, R. R. Martin, and T. Varady, *Algorithms for reverse engineering boundary representation models*, Computer Aided Design **33** (2001), no. 11, 839–851.

[Boc87] H.G. Bock, *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, Bonner Mathematische Schriften, vol. 183, Universität Bonn, Bonn, 1987.

[Boi84] J.-D. Boissonnat, *Geometric structures for three-dimensional shape representation*, ACM Transactions on Graphics **3** (1984), no. 4, 266–286.

[Bon95] A. le Bonheur, *Of gods, kings, and men. bas-reliefs of angkor wat and bayon*, Serindia Publishing, London, 1995.

[BS97] G. Barequet and M. Sharir, *Partial surface and volume matching in three dimensions*, IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997), no. 9, 929–948.

[BT95] P. T. Boggs and J. W. Tolle, *Sequential quadratic programming*, Acta numerica **4** (1995), 1–51, Cambridge Univ Press.

[BTFN+08] B. Brown, C. Toler-Franklin, D. Nehab, M. Burns, and D. Dobkin, *A system for high-volume acquisition and matching of fresco fragments: Reassembling theran wall paintings*, ACM Transactions on Graphics (TOG) **27** (2008), no. 3, 84.

[BW87] J.O. Berkey and P. Y. Wang, *Two-dimensional finite bin-packing algorithms*, The Journal of the Operational Research Society **38** (1987), no. 5, 423–429.

[CB05] O. Cunin and S. Baku, *The face towers at banteay chhmar*, GOTO Shoin Publishing CO Ltd, 2005.

[CB09] H. Chen and B. Bahnu, *Efficient recognition of highly similar 3d objects in range images*, IEEE IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009), no. 1, 172–179.

[CDV08] G. Carra, S. D'Amelio, and B. Villa, *The virtual reconstruction of temple b in selinunte excavation site*, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **37** (2008), no. 5, 391–396.

[CGK03] A. Cardone, S. K. Gupta, and M. Karnik, *A survey of shape similarity assessment algorithms for product design and manufacturing applications*, Journal of Computing and Information Science in Engineering **3** (2003), no. 2, 109–118.

[Chu08] C. Chu, *Electronic design automation*, ch. 11, Placement, pp. 635–685, Elsevier/Morgan Kaufmann, 2008.

[CJGJ96] E. G. Coffman Jr, M. R. Garey, and D. S. Johnson, *Approximation algorithms for bin packing: A survey*, Approximation algorithms for NP-hard problems **PWS Publishing Co.** (1996), 46–93.

[CL90] H. J. Conway and J. C. Lagarias, *Tiling with polyominoes and combinatorial group theory*, Journal of combinatorial theory **A 53** (1990), 183 – 208.

[DD92] K. A. Dowsland and W. B. Dowsland, *Packing problems*, European Journal of Operational Research **56** (1992), no. 1, 2–14.

[DeG71] H. DeGast, *Three dimensional puzzles*, Google Patents, May, 11 1971, US Patent 3,578,331.

[DFM+08] K. Dylla, B. Frischer, P. Mueller, A. Ulmer, and S. Haegler, *Rome reborn 2.0: A case study of virtual city reconstruction using procedural modeling techniques*, Computer Graphics World **16** (2008), 25.

[Die06] R. Diestel, *Graphentheorie*, 3. ed., Springer-Verlag, 2006.

[DIHOS07] J. Daniels II, L. K. Ha, T. Ochotta, and C. T. Silva, *Robust smooth feature extraction from point clouds*, IEEE International Conference on Shape Modeling and Applications, 2007. SMI '07., 2007, pp. 123–136.

[DIOHS08] J. Daniels II, T. Ochotta, L. K. Ha, and C. T. Silva, *Spline-based feature curves from point-sampled geometry*, Visual Computing **24** (2008), 449–462.

[DR01] J. Dumarcay and P. Royère, *Cambodian architecture. eigth to thirteenth centuries*, Handbook of Oriental Studies. Section Three. South-East Asia, vol. 12, Brill, 2001.

[Dra12] P. Drap, *Underwater photogrammetry for archaeology*, Special applications of photogrammetry, http://www. intechopen. com/books/special-applications-of-photogrammetry/underwater-photogrammetry-for-archaeology (2012), 111–136.

[DSS+07] P Drap, J Seinturier, D Scaradozzi, P Gambogi, L Long, and F Gauch, *Photogrammetry for virtual exploration of underwater archeological sites*, Proceedings of the 21st International Symposium, CIPA 2007: AntiCIPAting the Future of the Cultural Past: Athens (Greece), 01–06 October 2007, Citeseer, 2007.

[Dum73] J. Dumarcay, *Histoire architecturale du temple*, Bayon, vol. III, Mémoires archéologiques, no. 2, Publications de l'école francaise d'extrême-orient, 1973.

[DVVR06] K. Demarsin, D. Vanderstraeten, T. Volodine, and D. Roose, *Detection of closed sharp feature lines in point clouds for reverse engineering applications*, Geometric Modeling and Processing-GMP, Lecture Notes in Computer Science, vol. 4077, Springer, 2006, pp. 571–577.

[Dyc90] H. Dyckhoff, *A typology of cutting and packing problems*, European Journal of Operational Research **44** (1990), 145–159.

[EB08] S. Eliuk and P. Boulanger, *Algorithmic reconstruction of broken fragments*, Proceedings of the 36 th CAA Conference (2008).

[Ebe07] D. H. Eberly, *3d game enginge design*, 2nd ed., Morgen Kaufman Publisher, Elsevier, 2007.

[Eri05] C. Ericson, *Real time collision detection*, 2005.

[FC06] M. Freeman and J. Claude, *Ancient angkor*, Amarin Printing and Publishing Co. Ltd., 2006.

[FCOS05] S. Fleishman, D. Cohen-Or, and C. T. Silva, *Robust moving least-squares fitting with sharp features*, ACM Transactions on Graphics **24** (2005), no. 3, 544–552.

[FDF+97] J. D. Foley, A. v. Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips, *Introduction to computer graphics*, Addison-Wesley, 1997.

[FF13] B. Frischer and J. Fillwalk, *A computer simulation to test the buchner thesis. the relationship of the ara pacis and meridian in the campus martius, rome*, Digital Heritage International Congress, The Eurographics Association, 2013, pp. 341–345.

[FG64] Herbert Freeman and L Garder, *Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition*, Electronic Computers, IEEE Transactions on **IEEE** (1964), no. 2, 118–127.

[FK04] T. Funkhouser and M. Kazhdan, *Shape-based retrieval and analysis of 3d models*, ACM SIGGRAPH 2004 Course Notes, ACM, 2004, p. 16.

[Flo10] S. Floery, *Constrained matching of point clouds and surfaces*, Ph.D. thesis, Technische Universität Wien, 2010.

[For09] M. Fornasier, *Mathematics enters the picture*, Mathknow, Springer, 2009, pp. 217–228.

[FR91] T. MJ. Fruchterman and E. M. Reingold, *Graph drawing by force-directed placement*, Software: Practice and experience **21** (1991), no. 11, 1129–1164.

[FSN+11] J. Freudenreich, A. Schäfer, P. Nguonphan, M., and Hans Georg Bock, *Close range architecture documentation of angkor style temples*, Proceedings of DMACH 2011 Conference (Amman, Jordan), March, 13-15 2011.

[FTFC+11] T. Funkhouser, C. Toler-Franklin, A. G. Castaneda, B. Brown, D. Dobkin, S. Rusinkiewicz, and T. Weyrich, *Learning how to match fresco fragments*, Journal on Computing and Cultural Heritage (JOCCH) **4** (2011), no. 2, 7.

[GBS14] L. Gomes, E. R. P. Bellon, and L. Silva, *3d reconstruction methods for digital preservation of cultural heritage: a survey*, Pattern Recognition Letters (2014).

[GG07] G. Guennebaud and M. Gross, *Algebraic point set surfaces*, ACM Transactions on Graphics **26** (2007), no. 3, 23–1–23–9.

[GGG08] G. Guennebaud, M. Germann, and M. Gross, *Dynamic sampling and rendering of algebraic point set surfaces*, Computer Graphics Forum **27** (2008), no. 3, 653–662.

[GH09] A. Geary and E. Howe, *Three-dimensional documentation and virtual restoration of the lichfield angel*, Journal of the Institute of Conservation **32** (2009), no. 2, 165–179.

[GJS74] M. R. Garey, D. S. Johnson, and L. Stockmeyer, *Some simplified np-complete problems*, Proceedings of the sixth annual ACM symposium on Theory of computing, ACM, 1974, pp. 47–63.

[Gol70] S. W. Golomb, *Tiling with sets of polyominoes*, Journal of combinatorial theory **9** (1970), 60–71.

[Gro37] G. Groslier, *Une merveilleuse cité khmere: 'bantéai chmar' ville ancienne du cambodge*, L'Illustration **4909** (1937), no. 3, 352–257.

[GT06] H. Gross and U. Thoennessen, *Extraction of lines from laser point clouds*, Symposium of ISPRS Commission III: Photogrammetric Computer Vision PCV06. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 36, 2006, pp. 86–91.

[Gut09] A. J. Guttmann (ed.), *Polygons, polyominoes and polycubes*, Springer, 2009.

[GWM01] S. Gumhold, X. Wang, and R. MacLeod, *Feature extraction from point clouds*, Proceedings of 10th international meshing roundtable, 2001.

[HFG+06] Q. X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann, *Reassembling fractured objects by geometric matching*, Proceedings of ACM SIGGRAPH **25** (2006), no. 3, 569–578.

[HFR10] S. Hu, J. Feng, and M. Rasi, *Autonomous generation of bounding boxes for image sets of the same object*, 2010.

[Hig01] C. Higham, *The civilization of angkor*, University of California Press, 2001.

[Hig04] C.F. Higham, *Encyclopedia of ancient asian civilazations*, Facts on File Inc., 2004.

[HM13] K. J. Holyoak and R. G. Morrison, *The oxford handbook of thinking and reasoning*, Oxford University Press, 2013.

[IOT+07] K. Ikeuchi, T. Oishi, J. Takamatsu, R. Sagawa, A. Nakazawa, R. Kurazume, K. Nishino, M. Kamakura, and Y. Okamoto, *The great buddha project: digitally archiving, restoring, and analyzing cultural heritage objects*, International Journal of Computer Vision **75** (2007), no. 1, 189–208.

[JHG99] B Jaehne, H Haussecker, and P. Geissler (eds.), *Handbook of computer vision and applications*, vol. 1 Sensors and Imaging, Academic Press, 1999.

[Jun13] Dieter Jungnickel, *Graphs, networks and algorithms*, 4. ed., Springer, 2013.

[Jyl10] J. Jylänki, *A thousand ways to pack the bin – a practical approach to two-dimensional rectangle bin packing*, retrived from http://clb. demon. fi/files/RectangleBinPack. pdf (2010).

[KHW91] E. Kishon, T. Hastie, and H. J. Wolfson, *3-d curve matching using splines*, Journal of Robotic Systems **8** (1991), no. 6, 723–743.

[KK01] W. Kong and B. B. Kimia, *On solving 2d and 3d puzzles using curve matching*, Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001, vol. 2, 2001, pp. II–583–II–590.

[KL06] D. Koller and M. Levoy, *Computer-aided reconstruction and new matches in the forma urbis romae*, Bullettino Della Commissione Archeologica Comunale di Roma **2** (2006).

[Kol08] D. R. Koller, *Virtual archaeology and computer-aided reconstruction of the severan marble plan*, Beyond Illustration: 2D and 3D Digital Technologies as Tools for Discovery in Archaeology, British Archaeological Reports International Series (2008), 125–134.

[KS03] M. Kampel and R. Sablatnig, *Virtual reconstruction of broken and unbroken pottery*, Proceedings of the Fourth International Conference on 3D Digital Imaging and Modeling (3DIM '03), 2003, pp. 318–325.

[KS04] ———, *3d puzzling of archaeological fragments*, Proceedings of the 9th Computer Vision Winter Workshop, vol. 2, Slovenian Pattern Recognition Society, 2004, pp. 31 – 40.

[KS09] F. Kleber and R. Sablatnig, *A survey of techniques for document and archaeology artefact reconstruction*, Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on, IEEE, 2009, pp. 1061–1065.

[KS10] ———, *Scientific puzzle solving: Current techniques and applications*, Computer Applications and Quantitative Methods in Archaeology (CAA). Making History Interactive **Computer Applications to Archaeology 2009 Williamsburg, Virginia, USA. March 22-26, 2009** (2010).

[Laj11] E. L. de Lajonquiere, *Inventaire descriptif des monuments du cambodge*, vol. 9, Inventaire archeologique de l'indo-chine, no. 3, Ecole Francaise d'Extreme Orient, Paris, 1911.

[Lon98] S. Loncaric, *A survey of shape analysis techniques*, Pattern Recognition **31** (1998), no. 8, 983–1001.

[Mar97] W. R. Marshall, *Packing rectangles with congruent polyominoes*, Journal of combinatorial theory **A 77** (1997), 181 – 192.

[Mar12] H. Mara, *Multi-scale integral invariants for robust character extraction from irregular polygon mesh data*, Ph.D. thesis, Heidelberg University, 2012.

[MDC90] J. Mukherjee, P. P. Das, and B. N. Chatterji, *An algorithm for the extraction of the wire frame structure of a three-dimensional object*, Pattern Recognition **23** (1990), no. 9, 999–1010.

[Mor06] M.E. Mortenson, *Geometric modeling*, Industrial Press Inc., 2006.

[MP98] E. Milios and E. G. M. Petrakis, *Efficient shape matching and retrieval at multiple scales*, Dept. Comput. Sci., York Univ., Toronto, Ont., Canada (1998).

[MRS10] N. Mellado, P. Reuter, and C. Schlick, *Semi-automatic geometry-driven reassembly of fractured archeological objects*, VAST 2010: The 11th International Symposium on Virtual Reality, Archaeology and Cultural Heritage., 2010.

[Ngu09] P. Nguonphan, *Computer modeling, analysis and visualization of angkor wat style temples in cambodia*, Ph.D. thesis, Heidelberg University, 2009.

[NW06] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed., Springer Verlag, Berlin Heidelberg New York, 2006.

[OBS04] Y. Ohtake, A. Belyaev, and H.P. Seidel, *Ridge-valley lines on meshes via implicit surface fitting*, ACM Transactions on Graphics (TOG) **23** (2004), no. 3, 609–612.

[Oet13] T. Oetelaar, *Reconstructing the baths of caracalla*, CAA 2012, 2013.

[ON11] G. Oxholm and K. Nishino, *Reassembling thin artifacts of unknown geometry*, Proceedings of the 12th International conference on Virtual Reality, Archaeology and Cultural Heritage, 2011, pp. 49–56.

[ON13] ———, *A flexible approach to reassembling thin artifacts of unknown geometry*, A flexible approach to reassembling thin artifacts of unknown geometry **14** (2013), no. 1, 51–61.

[O'R85] J. O'Rourke, *Finding minimal enclosing boxes*, International journal of computer & information sciences **14** (1985), no. 3, 183–199.

[PAE+12] C. Papaodysseus, D. Arabadjis, M. Exarhos, P. Rousopoulos, S. Zannos, M. Panagopoulos, and L. Papazoglou-Manioudaki, *Efficient solution to the 3d problem of automatic wall paintings reassembly*, Computers & Mathematics with Applications (2012).

[PKT01] G. Papaioannu, E.-A. Karabassi, and T. Theoharis, *Virtual archaeologist: Assembling the past*, Computer Graphics and Applications, IEEE **21** (2001), no. 2, 53 – 59.

[PPCS] G. Palmas, N. Pietroni, P. Cignoni, and R. Scopigno, *A practical framework for assembling fragmented objects*.

[PPE+02] C. Papaodysseus, T. Panagopoulos, M. Exarhos, C. Triantafillou, D. Fragoulis, and C. Doumas, *Contour-shape based reconstruction of fragmented, 1600 bc wall paintings*, Signal Processing, IEEE Transactions on **50** (2002), no. 6, 1277–1288.

[Pre00] D. Preston, *The temples of angkor. still under attack*, National Geographic **198** (2000), no. 2, 82–103.

[PSCC07] D. Parikh, R. Sukthankar, T. Chen, and M. Chen, *Feature-based part retrieval for interactive 3d reassembly*, Applications of Computer Vision, 2007. WACV'07. IEEE Workshop on, 2007, pp. 14–20.

[RL01] S. Rusinkiewicz and M. Levoy, *Efficient variants of the icp algorithm*, Third International Conference on 3-D Digital Imaging and Modeling (3DIM '01), 2001, pp. 145–152.

[Roo06] D. Rooney, *Angkor*, Odyssey, 2006.

[RRS+07] P. Reuter, G. Rivière, N. Sorraing, L. Espinasse, and R. Vergnieux, *Archeotui - a tangible user interface for the virtual reassembly of fractured archeological objects*, Proceedings of the 8th International conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage, Eurographics Association, 2007, pp. 15–22.

[SBG11] F. Stanco, S. Battiato, and G. Gallo, *Digital imaging for cultural heritage preservation: Analysis, restoration, and reconstruction of ancient artworks*, CRC Press, 2011.

[SBSL14] A. Schäfer, H. Bock, J. Sanday, and H. Leitte, *Virtually reassembling angkor-style khmer temples*, Digital Applications in Archaeology and Cultural Heritage (2014), submitted.

[Sch09] A. Schäfer, *On scaling techniques and termination criteria for SQP methods*, Diploma thesis, Universität Heidelberg, 2009.

[SDF+10] H. Shin, C. Doumas, T. Funkhouser, S. Rusinkiewicz, K. Steiglitz, A. Vlachopoulos, and T. Weyrich, *Analyzing fracture patterns in theran wall paintings*, The 11th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST (A. Artusi, M. Joly-Parvex, G. Lucet, A. Ribes, and D. Pitzalis, eds.), Eurographics Association, 2010, pp. 71–78.

[SHH99] S. Suri, P. M. Hubbard, and J. F. Hughes, *Analyzing bounding boxes for object intersection*, ACM Transactions on Graphics **18** (1999), no. 3, 257–277.

[SHI96] H.-Y. Shum, M.l Hebert, and K. Ikeuchi, *On 3d shape similarity*, Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on, IEEE, 1996, pp. 526–531.

[SJC] P. D. Sharrock, C. Jacques, and O. Cunin, *Banteay chhmar: Uncovering the last great forest temple of ancient cambodia*, River Books.

[SMF+10] A. Schaefer, H. Mara, J. Freudenreich, S. Kroemker, and H.G. Bock, *Visualization and documentation of weathered bas-reliefs using close-range 3d-scanners*, CIPA, accepted (2010).

[SMF+11] A. Schäfer, H. Mara, J. Freudenreich, B. Breuckmann, C. Düffort, and H. G. Bock, *Large scale angkor style reliefs: High definition 3d acquisition and improved visualization using local feature estimation*, Revive the Past, Proceedings of the 39th CAA Conference (2011), 70–80.

[SSRS06] T. Sonnemann, M. Sauerbier, F. Remondino, and G. Schrotter, *Reality-based 3d modeling of the angkorian temples using aerial images*, BAR INTERNATIONAL SERIES **1568** (2006), 573.

[TFBW+10] C. Toler-Franklin, B. Brown, T. Weyrich, T. Funkhouser, and S.N. Rusinkiewicz, *Multi-feature matching of fresco fragments*, ACM Transactions on Graphics (Proc. SIGGRAPH Asia), vol. 6, dec 2010.

[TNP11] E. Tsamoura, N. Nikolaidis, and I Pittas, *Digital reconstruction and mosaicing of cultural arifacts*, Digital Imaging for Cultural Heritage Preservation: Analysis, Restoration, and Reconstruction of Ancient Artworks (2011), 353.

[Tul05] J. Tully, *A short history of cambodia. from empire to survival*, Allen & Unwin, 2005.

[Tut63] W. T. Tutte, *How to draw a graph*, Proc. London Math. Soc. **13** (1963), no. 3, 743–768.

[Tyb04] R. Tybon, *Generating solutions to the jigsaw puzzle problem*, Ph.D. thesis, Griffith University, 2004.

[UT99] G. Ücoluk and H. I. Toroslu, *Automatic reconstruction of broken 3-d surface objects*, Computers and Graphics **23** (1999), no. 4, 573–582.

[VGSR04] G. Vosselman, B. G. Gorte, G. Sithole, and T. Rabbani, *Recognising structure in laser scanner point clouds*, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences **46** (2004), no. 8, 33–38.

[VMC97] T. Varady, R. R. Martin, and J. Cox, *Reverse engineering of geometric models - an introduction*, Computer-Aided Design (1997), 255–268.

[WC04] A. R. Willis and D. B. Cooper, *Bayesian assembly of 3d axially symmetric shapes from fragments*, Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 1, IEEE, 2004, pp. I–82.

[WC08] A.R. Willis and D.B. Cooper, *Computational reconstruction of ancient artifacts*, IEEE Signal Processing Magazine **25** (2008), no. 4, 65–83.

[WFRA99] N. Werghi, R. Fisher, C. Robertson, and A. Ashbrook, *Object reconstruction by incorporating geometric constraints in reverse engineering*, Computer Aided Design **31** (1999), no. 6, 363–399.

[Win06] S. Winkelbach, *Das 3d-puzzle-problem: effiziente methoden zum paarweisen zusammensetzen von dreidimensionalen fragmenten*, Shaker Verlag, 2006.

[WSKY88] H. Wolfson, E. Schonberg, A. Kalvin, and L. Yehezkel, *Solving jigsaw puzzles by computer*, Annals of Operations Research **12** (1988), 51 – 64.

[WW08] S. Winkelbach and F. Wahl, *Pairwise matching of 3d fragments using cluster trees*, International Journal of Computer Vision **78** (2008), no. 1, 1–13.