

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

vorgelegt von
Dipl.–Math. Silvia Harmsen
aus Ermelo (die Niederlande)

Tag der mündlichen Prüfung: 27. Oktober 2006

Algorithmic Computer Reconstructions of Stalactite Vaults - Muqarnas - in Islamic Architecture

Gutachter: Prof. Dr. Gerhard Reinelt

Prof. Dr. Jan P. Hogendijk

For Cor

Zusammenfassung

Muqarnas, oder Stalaktitengewölbe, sind dreidimensionale Ornamente, die in der islamischen Architektur verbreitung finden. Sie werden in Gewölben, Kuppeln, Nischen, auf Bogenkonstruktionen, oder als fast flache dekorative Friesen genutzt. Ein Muqarnas hat die Aufgabe, einen fließenden Übergang von einer geraden Wand zu einem gekrümmten Teil zu gewährleisten. Ein Muqarnasgewölbe wird von verschiedenen nischeartigen Elementen, die in horizontalen Stockwerken angeordnet sind, aufgebaut. Ein Hauptmerkmal der Muqarnas ist, dass die Form eine dreidimensionale Einheit darstellt, welche in einem zweidimensionalen Grundriss repräsentiert werden kann. Wir konzentrieren uns auf die Frage, ob die zweidimensionale Projektion alle Strukturinformationen des dreidimensionalen Gewölbes enthält.

Um einen Rahmen zu schaffen, in dem wir die Muqarnasstruktur beschreiben können, führen wir explizite Definitionen ein. Jedes Muqarnaselement wird parametrisiert, um Typ, Abmessungen und Position in dem Gewölbe beschreiben zu können. Ein graphentheoretischer Ansatz ermöglicht es, die Strukturinformationen des dreidimensionalen Muqarnas in einem zweidimensionalen Grundriss zu integrieren. Dies wird durch die Konstruktion gerichteter Teilgraphen aus dem Muqarnasentwurf realisiert. Die Hauptaufgabe ist es, alle gerichteten Teilgraphen eines Muqarnasentwurfs zu finden, für die eine dreidimensionale Muqarnasrepräsentation möglich ist. Die dreidimensionalen Computerrekonstruktionen können dann direkt von den gerichteten Teilgraphen ausgehend aufgebaut werden. Wir haben einen Algorithmus entwickelt zur Konstruktion der gerichteten Graphen und zur dreidimensionalen Rekonstruktion des Muqarnasgewölbes.

Der Algorithmus wurde in zwei Computerprogrammen implementiert. Das Programm `plantograph` findet alle gerichteten Teilgraphen, die mit einer Muqarnasstruktur eines gegebenen Muqarnasgrundrisses, korrespondieren. Das Programm `graphtomuq` übersetzt einen Teilgraph in eine dreidimensionale Computerrekonstruktion. Die verschiedenen Teilgraphen ergeben verschiedene Muqarnasrekonstruktionen mit gleicher Projektion. Eine Interaktion mit dem Anwender ist nötig um die gewünschte Rekonstruktion auszuwählen, wofür Fachwissen in Kunstgeschichte erforderlich ist.

Die Unterschiede zwischen Muqarnasgewölben mit gleicher Projektion können *lokal* oder *global* sein. Lokal bedeutet, dass die Muqarnas durch Austausch einzelner Elemente ineinander überführt werden können. Muqarnas sind global unterschiedlich, wenn mehrere Elemente miteinbezogen sind und dadurch die geometrische Form und die Anzahl der Stockwerke variiert. Die Rekonstruktionsmöglichkeiten können eingeschränkt werden, wenn die Abmessungen des Gewölbes, in das ein Muqarnas eingebaut werden soll, bekannt sind. Die Anzahl der Stockwerke ist dadurch beschränkt.

Wir konzentrieren uns auf Muqarnasgewölbe, die in Kuppeln oder Nischen eingebaut sind und der seldschukischen oder ilkhanidischen Zeitperiode entstammen. Der Algorithmus wurde an verschiedenen Muqarnasgrundrissen aus Anatolien und dem Iran getestet. Zudem wird eine neue mögliche Interpretation für den ältesten bekannten Muqarnasentwurf, eine in Takht-i-Sulaymān gefundene Platte, präsentiert.

Abstract

Muqarnas, or stalactite vaults, are three-dimensional ornaments, common in the Islamic architecture. They are used in vaults, domes, niches, arches, and as an almost flat decorative frieze. It is the function of a muqarnas to guarantee a smooth transition between straight walls and more curved parts. A muqarnas vault is built from different niche-like elements, arranged in horizontal tiers. One of the main characteristics of muqarnas is its form as a three-dimensional unit that can be represented as a two-dimensional outline. In this study, we focus on the question whether this two-dimensional projection contains all structure information of the three-dimensional muqarnas vault.

Explicit definitions are given to create a framework in which we are able to describe the muqarnas structure. Each muqarnas element is described by means of parameters representing its type and measurements together with parameters that describe the position of the element in the muqarnas vault. A graph theoretical approach makes it possible to include the structure information of the three-dimensional muqarnas in the two-dimensional outline. This is done by constructing a directed subgraph from the muqarnas design. The main task is then to find all directed subgraphs corresponding to a muqarnas design for which a three-dimensional muqarnas representation is possible. We can construct three-dimensional computer reconstructions directly from the directed subgraphs. An algorithm is developed for constructing the directed subgraphs and reconstructing the three-dimensional muqarnas.

Two software tools are designed to execute this process. The program `plantograph` finds the directed subgraphs corresponding to a muqarnas structure from a muqarnas design. The program `graphtomuq` converts a subgraph into a three-dimensional computer reconstruction of the muqarnas. The various subgraphs result in different muqarnas reconstructions with the same plane projection. Some interaction is necessary to select the required reconstruction, for this art historical expertise is essential.

The differences between muqarnas with the same plane projection can be *local* or *global*. Local means that by exchanging several elements the muqarnas are equal. Muqarnas are globally different if more elements are involved and the shape, or number of tiers, of the muqarnas differ. We can restrict the number of reconstructions if we know the proportions of the vault into which the muqarnas needs to fit. This gives a restriction on the amount of tiers that can be used.

This work focuses on the muqarnas vaults fitting into domes or niches from the Seljuk and Il-Khanid periods. Different tests of such muqarnas from Anatolia and Iran are given. Additionally a new interpretation for the oldest known design which is found at Takhti-Sulaymān is presented.

Contents

Zusammenfassung	i
Abstract	iii
1 Introduction	1
1.1 Muqarnas in the Islamic Architecture	1
1.1.1 Overview about Seljuk and Il-Khanid Architecture	1
1.1.2 What is a Muqarnas?	2
1.1.3 Muqarnas Designs	5
1.2 Purpose and Contribution of our Research	6
1.2.1 Motivation	7
1.2.2 Literature Overview	8
1.2.3 Outline of this Work	9
2 Structure of the Muqarnas	11
2.1 Muqarnas in the Architecture	11
2.1.1 Muqarnas Elements	11
2.1.2 Three-Dimensional Muqarnas Structure	16
2.2 Muqarnas as an Abstract Geometrical Structure	19
2.2.1 Elements as Abstract Objects	19
2.2.2 Muqarnas Plan	21
2.2.3 Structure of Il-Khanid Muqarnas	26
3 Algorithm for Reconstructing Muqarnas	31
3.1 Representation of the Muqarnas Structure in a Graph	31
3.1.1 Definition of the Muqarnas Graph	32
3.1.2 Properties of the Muqarnas Graph	33
3.2 Construction of Muqarnas Graphs from the Plan	39
3.3 Conversion of Muqarnas Graph into Muqarnas Structure	45
3.3.1 Muqarnas Reconstruction Process	45
3.3.2 Determination of Faces from the Graph	47
3.3.3 Conversion of Faces into Muqarnas Elements	48
3.3.4 Reconstruction from a Simplified Plan	49

3.4	Preparation of the Plan	51
3.5	Uniqueness of the Reconstructions	55
4	Software Tools for Reconstructing Muqarnas	57
4.1	The Program <code>plantograph</code>	57
4.1.1	Conversion of the Plan into the Graph	58
4.1.2	Representation of the Muqarnas Plan	60
4.1.3	Conversion of the Input into a Muqarnas Plan	61
4.1.4	Boundary of a Muqarnas Plan	64
4.1.5	Determination of the Direction of the Edges	67
4.1.6	Complexity Analysis	69
4.2	The Program <code>removelines</code>	71
4.2.1	Removal of Edges from the Plan	71
4.2.2	Complexity Analysis	73
4.3	The Program <code>graphtomuq</code>	73
4.3.1	Calculation of the Muqarnas Structure	73
4.3.2	Visualization of the Muqarnas in VRML	76
4.3.3	Visualization of the Elements	77
4.3.4	Complexity Analysis	82
5	Results of Algorithmic Muqarnas Reconstructions	83
5.1	Computer Reconstructions of Seljuk Muqarnas	84
5.1.1	Alay Han	84
5.1.2	Kayseri, Çifte Medrese	86
5.1.3	Sultan Han near Kayseri	89
5.1.4	Ankara, Arslanhane Camii	92
5.2	Computer Reconstructions of Il-Khanid Muqarnas	95
5.2.1	Natanz, Friday Mosque	95
5.2.2	Biṣṭām, Shrine of Bāyazīd	98
5.2.3	Farūmad, Friday Mosque	100
5.2.4	Takht-i-Sulaymān, South Octagon	103
5.2.5	Takht-i-Sulaymān, Plate	105
6	Conclusions	113
A	Graph Theoretical Terminology	117
	Acknowledgements	121

Chapter 1

Introduction

In this work we study *muqarnas*, a special three-dimensional ornamentation, common in the Islamic architecture. The aim is to analyze the two-dimensional designs of muqarnas and to create three-dimensional computer reconstructions of the muqarnas directly from their designs. Before going into details, we will use this introductory chapter to explain what muqarnas are. We focus on muqarnas from the Seljuk and Il-Khanid style. In Section 1.1 a short overview of the history of the Seljuk and Il-Khanid time periods is given. After that, muqarnas and their designs are explained. In Section 1.2 the motivation of this work and problem formulation is presented. Furthermore, a very short summary of previous research works concerning muqarnas is included. We finish this chapter with an outline of the research contained in this work.

1.1 Muqarnas in the Islamic Architecture

In this work we study muqarnas built in Seljuk and Il-Khanid times, the periods in which they got important in Islamic architecture. We therefore start in Section 1.1.1 with a historical overview of these times. After that, we will use in Section 1.1.2 the definitions from al-Kāshī, a mathematician from Timurid time, to explain what muqarnas are. Finally, we discuss in Section 1.1.3 the aspects of muqarnas designs, illustrated by a famous design found at the Takht-i-Sulaymān.

1.1.1 Overview about Seljuk and Il-Khanid Architecture

The word Seljuk is used to denote a mighty group of nomadic Turks, who invaded a large part of Asia (see e.g. [Aslanapa, 1971]). Seljuk was the name of one of their first supervisors. Under his supervision a tribe of the Oghuz Turks converted to Islam around 960. They ruled parts of central Asia and Middle East from the 11th to 14th century. The Great Seljuk Empire reached its peak during the reigns, in the time between 1038 and 1093, of the three sultans Tughrul Bek, grandson of Seljuk, Alp Arslan and Malikshāh. Under Malikshāh and his two Persian viziers Nizām al-Mulk and Tāj al-Mulk, the Seljuk

state expanded in various directions to Persian border, so that it bordered China in the East and the Byzantine in the West. After the death of Malikshāh the empire split into smaller parts. Ahmed Sanjar one of the sons of Malikshāh was the ruler of most of Persia. This part, the so called Seljuk Dynasty, collapsed after his death in 1157. In Anatolia, for example, the Seljuks ruled longer. They are called the Seljuks of Rūm and reigned until the Mongol invasion in 1307.

From 1220 on, there were different successful attacks on Central Asia by the Mongols under supervision of Genghis Khan (see e.g. [Wilber, 1955]). In 1256 his grandson Hūlagu Khan invaded Baghdad and founded the Il-Khanid state. During the leadership of Hūlagu's great-grandson Ghāzān Khan the Il-Khanids converted to Islam and the culture flourished. After the death of the last Khan, Abu Sa'id, in 1335 the Il-Khanids lost their power and the state broke up into different dominions, which developed separately.

By the diffusion of the Islamic religion during the Seljuk and Il-Khanid periods a lot of religious buildings, like mosques and medreses, arose. The mosques are the places where Muslims come together for their Friday prayer. They became the social centers of the city. The first mosques from the seventh century were open-air spaces. During Seljuk and Il-Khanid time they mainly consist of a quadratic room covered by a dome.

Medreses are educational establishments, nowadays also known as Koran Schools. The vizier Nizām al-Mulk developed a political system where jurisprudence became very important. Besides of theology, in the medreses, law, political science, languages, literature and science was taught. The medreses transformed into an extensive state organization of institutions.

An example of non-religious buildings, which raised during Seljuk and Il-Khanid times, were the caravanserais (see e.g. [Erdmann, 1961]). A caravanseray is a roadside inn which was important as a place for the merchants and the animals to rest and recover from their journey. Often the caravanserais contained shops where the merchants could dispose some of their goods. Caravanserais supported the flow of commerce, information and people across a network of trade routes of Asia, North Africa and South-Eastern Europe. Most typically it was a building with a square or rectangular walled exterior, with a single portal wide enough to permit large or heavily laden animals to enter. The courtyard is almost always open to the sky, and along the inside walls of the enclosure we find several identical stalls, niches or chambers to accommodate merchants, their servants, animals and merchandise.

1.1.2 What is a Muqarnas?

Muqarnas is the Arabic word for stalactite vault, an architectural ornament consisting of niche-like elements, arranged in tiers. It can be used in domes, niches, or on arches as an almost decorative frieze. In Figure 1.1 different examples of muqarnas are shown. A *dome* is a half sphere, or an oval, usually placed on a rectangular building. A muqarnas can be fit into the dome to provide a smooth transition between the rectangular part and the vaulted part (see e.g. [al Asad, 1994, p.65] or [Bianca, 2000, p.45]). Figure 1.1(d)

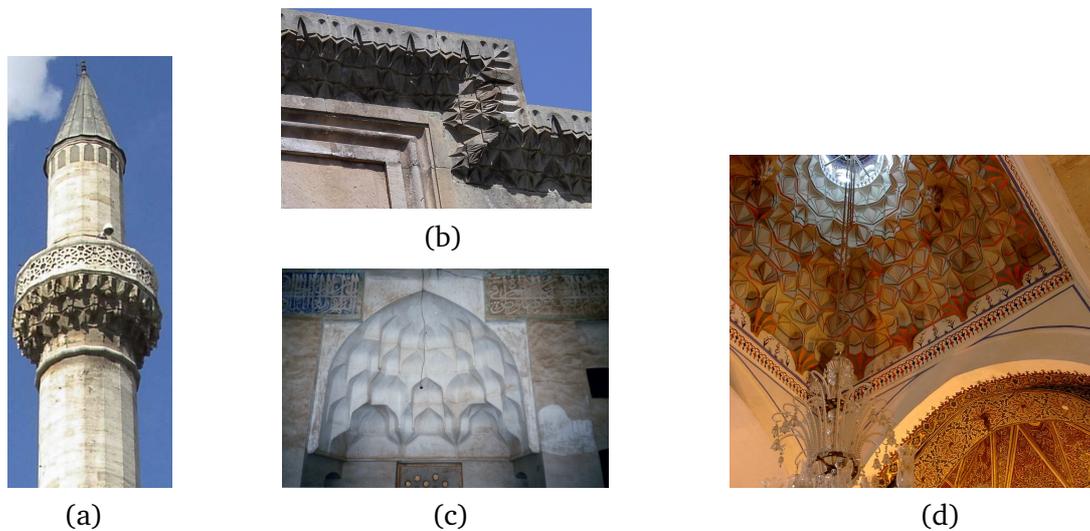


Figure 1.1. (a): Minaret of the Selimiya Camii in Konya (Turkey) decorated with muqarnas. (b): Cornice at the Sultan Han near Kayseri (Turkey) ornamented with muqarnas. (c): Muqarnas vault above a niche in the Friday Mosque in Natanz (Iran). (d): Muqarnas in a dome in the Mevlana Teke in Konya (Turkey).

shows an example of a muqarnas fit into a dome in the Mevlana Teke in Konya. Other famous examples of muqarnas in domes can be found in the Alhambra in Granada (see e.g. [Grabar, 1992] or [Jones and Goury, 2001]). In a niche a muqarnas is also used to hide the transition between the straight and curved parts of the niche. Niches are often found at entrance portals of Islamic buildings, for example at mosques. In Figure 1.1(c) we find an example of a muqarnas in a niche, it shows the basement vault of the north iwan from the Friday Mosque in Natanz. Examples of muqarnas as a decorative frieze can be found at balconies of minarets (see Figure 1.1(a)), towers which are attached to a mosque, often used for the call to prayer. Cornices, the uppermost section of moldings along the top of a wall or just below a roof, are also often decorated with muqarnas (see Figure 1.1(b)).

The fifteenth century mathematician Ghiyāth al-Dīn Mas'ūd al-Kāshī (died in 1429) defines the muqarnas in his *'Key of Arithmetic'* (see [Dold-Samplonius, 1992, p. 226]) from a practical point of view:

The muqarnas is a roofed (musaqqaf) [vault] like a staircase (madraj) with facets (ḍil') and a flat roof (saṭh). Every facet intersects the adjacent one at either a right angle, or half a right angle, or their sum, or another combination of these two. The two facets can be thought of as standing on a plane parallel to the horizon. Above them is built either a flat surface not parallel to the horizon, or two surfaces, either flat or curved, that constitute their roof. Both facets together with their roof are called one cell (bayt). Adjacent cells, which have their bases on one and the same surface parallel to the horizon, are

called one tier (*ṭabaqa*). The measure of the base of the largest facet is called the module (*miqyās*) of the muqarnas.

In addition there are intermediate elements which connect the roofs of adjacent cells or other intermediate elements. A more comprehensive explanation about the muqarnas elements will be given in Chapter 2. Al-Kāshī describes the plane projection of an element (either cell or intermediate element) as a basic geometrical form, namely, a square, half-square (cut along the diagonal), rhombus, half-rhombus (isosceles triangle having as base the shorter diagonal of the rhombus), almond (deltoid), jug (quarter octagon), large biped (complement to a jug), and small biped (complement to an almond). These elements are constructed according to the same unit of measure and fit together in a wide variety of combinations. Al-Kāshī uses in his computation the module of the muqarnas, defined as the base of the largest facet (the side of the square) as a basis for all proportions. Different materials can be used to construct muqarnas. For example, plaster, brick or wood are common materials. In Seljuk architecture brick is often used, in Il-Khanid architecture plaster is more common. Dependent of the material used for the muqarnas elements, there can be a wider variety in the elements. If the muqarnas is not constructed from prefabricated elements, we only need to care that two joining elements need to fit together. In other parts of the muqarnas the elements may have other measurements.

When a muqarnas structure has to be inserted into an existing vault, the height of the facets of the elements need to be adapted. The height of the facets often decreases in the upper tiers as the curvature of the vault diminishes. The part above the last tier, can then be finished in several ways. In some vaults the original brick work is left visible, in others the ceiling is plastered and ornamented by painting, or the upper part is filled with barley-kernels (see Section 2.1.1).



Figure 1.2. Muqarnas elements as found at the Takht-i-Sulaymān. On the left: cells. On the right: combinations of intermediate elements.

Examples of muqarnas fitting well in the definitions of al-Kāshī are the muqarnas remains found at the Takht-i-Sulaymān. The Takht-i-Sulaymān is an Il-Khanid seasonal palace in the Azerbaijan region of Iran, two hundred miles south of Tabriz and south-east of Lake Urmia. This palace is built over a Sasanian fire temple that seems to have been in use from the late fifth to the early seventh century, and continued to serve as a Zoroastrian sanctuary for two more centuries after Persia had been conquered by Islam.

Buried under the ruins of the palace several muqarnas remains were found constructed from probably prefabricated muqarnas elements (see Figure 1.2). These could have been used to construct several different vaults within the palace, but where such vaults might have been located can no longer be determined. Concerning the excavated muqarnas elements, described in detail by Harb (see [Harb, 1978]), we note that they have been found in three different sizes, namely, with units of measurement 21, 26, and 42 cm. Only one element of 42 cm (twice 21) has been found, and its height is undetermined, as it is an intermediate element without facet. The height of the cells with a unit of measurement of 21 is 42, and the height of the cells with a unit of measurement of 26 is 52. Hence, the height of the cells is twice their unit of measurement. The elements with a unit of measurement of 21 might originate from a muqarnas in the south octagon.

1.1.3 Muqarnas Designs

A muqarnas is usually designed by drawing the plane projection of the muqarnas elements. The first known example of a muqarnas design is a stucco plate found at Takht-i-Sulaymān with a muqarnas design incised in it (see [Harb, 1978]). Another famous example of a collection of designs is the Topkapı Scroll (see [Necipoğlu, 1995]). Until Necipoğlu's discovery of the Topkapı Scroll, the earliest known examples of such architectural drawings were a collection of fragmentary post-Timurid design scrolls on sixteenth-century Samarkand paper, retained at the Uzbek Academy of Sciences in Tashkent (see [Notkin, 1995]). These scrolls almost certainly reflect the ambitious Timurid drafting methods of the fifteenth century. The Timurid and post-Timurid scrolls show a decisive switch to more complex muqarnas with an increasing variety of polygons and star polygons.

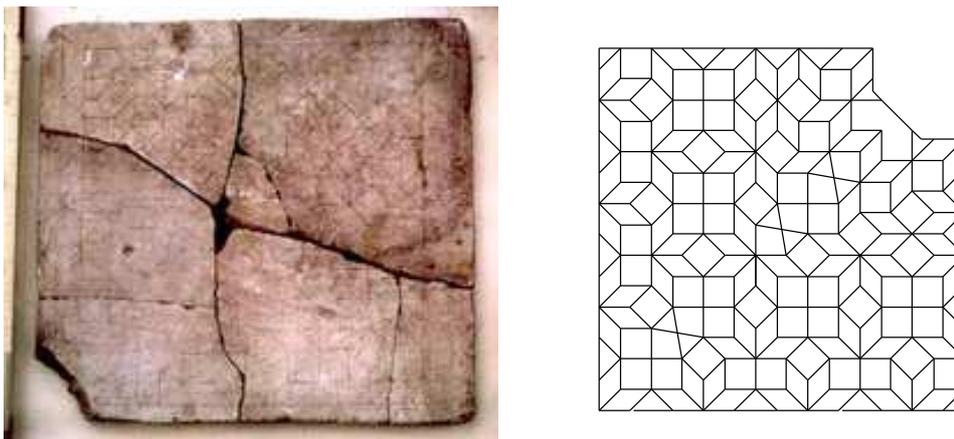


Figure 1.3. On the left: picture of the plate found at Takht-i-Sulaymān with a muqarnas design incised in it. On the right: the lines of the plate as recognized by Harb (after [Harb, 1978]).

The first study of the plate found at the Takht-i-Sulaymān is done by Harb, one of

the members of the excavation team who found this plate (see [Harb, 1978]). More recent studies and also suggestions to decode the plate can be found in [Yaghan, 2000] and [Dold-Samplonius and Harmsen, 2005]. The design on the plate (see Figure 1.3 on the left) is a about 3.5 to 4 cm thick rectangular plate with height 47 cm and width 50 cm. It contains a geometric grid in a quadratic field of 42 cm length. A small part of the bottom left corner is broken off and did not survive. The remaining plate is broken into seven parts, which fit together except near the middle of the plate, where there is a small hole. Under the clearly drawn grid are poorly erased lines visible. At some points these may have been auxiliary lines, but at others seem to have no direct connection with the actual design. The artisan seems to have reworked his design, and it is not clear whether the present design was ever used, or whether it was altogether abandoned. In the right picture of Figure 1.3 the lines of the plate as recognized by Harb are shown. As there were more muqarnas in the palace at the Takht-i-Sulaymān and because we do not know whether the design was ever used, the prefabricated elements found at the Takht-i-Sulaymān do not necessarily correspond to the plate. Nowadays the plate is kept in the Islamic Department of the Iran Bastan Museum in Tehran.

The design consists mainly of squares and rhombi, with isosceles right triangles along the frame of the field. The sides of the squares and rhombi, as well as the legs of the triangles, are all 3.5 cm in length. Their areas have been symmetrically arranged around a diagonal axis. The construction is completed in the upper right corner by an irregular quarter octagon. The angles of the various figures shapes are all multiples of 45° , with the exception of some semi-regular quadrangles and isosceles triangles along the diagonal.

The Topkapı Scroll, is a late fifteenth- or early sixteenth-century scroll containing patterns for ornamentations and patterns to be used as designs for muqarnas. This scroll is preserved at the Topkapı Museum in Istanbul and published by Gülru Necipoğlu (see [Necipoğlu, 1995]). It includes several rough designs of which the artist has shown a small part worked out in detail, probably to avoid confusion. It ranks among the oldest extant designs for muqarnas.

1.2 Purpose and Contribution of our Research

Our research differs from previous works about muqarnas. We do not only concern about a better understanding of the muqarnas, but also give an algorithmic way to build them directly from their design. We give in Section 1.2.1 the motivation of this study and the problem formulation. Then we summarize in Section 1.2.2 the contents of previous research works concerning muqarnas. The final part, Section 1.2.3, contains an outline of this work.

1.2.1 Motivation

This research focuses on the connection between a three-dimensional muqarnas structure and its two-dimensional plane projection. In the plane projection the muqarnas elements do not overlap, which makes it possible to design a muqarnas by drawing its plane projection. Nevertheless, such two-dimensional designs do not include explicitly all the three-dimensional information. A muqarnas consists of different niche-like elements, arranged in tiers. In the drawing we can distinguish the different elements, but we do not have information to which tiers the elements belong. It is also not directly clear which side of a certain polygon in the projection corresponds to the front part of the element and thus the orientation of the elements cannot be recognized. We have furthermore to take into account that there exist different elements with the same plane projection.

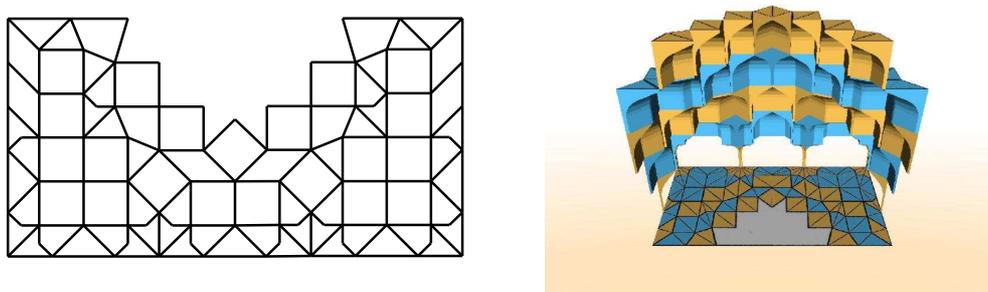


Figure 1.4. On the left: plane projection of the muqarnas vault in the basement of the north iwan of the Friday Mosque in Natanz. On the right: computer reconstruction of the muqarnas vault.

In this thesis we want to analyze the plane projection of muqarnas and study which part of the “lost” information can be calculated from the plane projection. The aim of this work is to answer the following questions:

- *Is there a direct algorithm for constructing a three-dimensional muqarnas from its plane projection?*
- *Are these muqarnas uniquely reconstructible?*

We restrict ourselves to muqarnas used in domes and niches, which are built during Seljuk or Il-Khanid times. This delivers an insight in the development of the oldest muqarnas forms. The purpose of our reconstruction algorithm is to analyze the *structure* of the muqarnas. It answers the question to what kind of element a polygon in the design corresponds and it determines its position (location and orientation). However, it does not tell us what style (colors, material) of elements need to be used.

The only known design of an Il-Khanid muqarnas is the plate found at Takht-i-Sulaymān. We do not have a real muqarnas corresponding to this plate, nor do we have

measures of the dome the muqarnas could fit in. Therefore, we can not verify our reconstructions. For Seljuk muqarnas we do not even have an example of a design. In this study we therefore work with plane projections of existing muqarnas, instead of working with original designs.

Our goal is to write a computer program which is able to propose a three-dimensional computer reconstruction for a given plane projection of a muqarnas. If we are able to write such a program, we can confirm our first question. The final software product should show all possible reconstructions corresponding to a certain plane projection so that we can answer the second research question. This software then can be useful during restoration and for designing new muqarnas. It gives the possibility to test different alternatives for a muqarnas reconstruction with a given plane projection.

1.2.2 Literature Overview

Our work is mainly inspired by a chapter of the manuscript “Key of Arithmetic”, written by the Timurid astronomer and mathematician al-Kāshī (see [al Kāshī, 1558]) and a research about Il-Khanid muqarnas of Harb (see [Harb, 1978]).

Al-Kāshī ranks among the greatest mathematicians and astronomers in the Islamic world. Two years before he died in 1429, he had finished the “Key of Arithmetic (Miftāḥ al-Ḥisāb)”, one of his major works, which he intended for everyday use. By far the most extensive part of the work is Book IV, “On Measurements”. The last section of Chapter 9, “Measuring Structures and Buildings”, is about muqarnas. The aim of al-Kāshī is to measure the muqarnas. He does not give us information about the construction of muqarnas and his work does not contain information about how we need to interpret muqarnas designs. For measuring the muqarnas, al-Kāshī gives an overview of the different muqarnas elements. He describes the surfaces of the elements by means of angles and circle arches. This makes it possible to give a quite exact calculation of the surface area. In this thesis we work with the translation, “Measuring the Muqarnas by al-Kāshī” of Yvonne Dold-Samplonius (see [Dold-Samplonius, 1992]).

Ulrich Harb was one of the members of the German excavation team who discovered the plate found at the Takht-i-Sulaymān. The aim of his work is to understand this muqarnas design. In his research, he gives an exact description of the muqarnas elements and he studies several still existing Il-Khanid muqarnas vaults. He concludes with an interpretation of the design on the plate. A quarter of his interpretation is presented as a pen drawing.

More recent research about muqarnas is done by Mohammad A. J. Yaghan. In his thesis, Yaghan gives a review of the terms used in the literature to describe muqarnas (see [Yaghan, 2001a]). The origin of muqarnas is also discussed and a computer algorithm to generate muqarnas is introduced. This muqarnas generation system is restricted to create one type of muqarnas only. The generated muqarnas are constructed in a radial way and every tier consists of one type of repeated elements. The system is not able to generate muqarnas which consists of varying element combinations, like common in Seljuk and Il-

Khanid architecture. Besides this work, Yaghan studies in [Yaghan, 2001b] the muqarnas elements in order to design new elements for creating new kind of muqarnas. Studies of Yaghan concerning the decoding of muqarnas designs contain a survey about corbelled muqarnas (see [Yaghan, 2003]) and also a reconstruction proposal for the plate found at Takht-i-Sulaymān (see [Yaghan, 2000]).

The thesis of Sayed M. A. Ibrahim documents the architectural features of muqarnas (see [Ibrahim, 2002]). This documentation serves to show how to implement repair work at historical buildings as well as to show how to integrate this traditional element into modern architecture. It is a complete overview, which is very useful to understand muqarnas. Both Yaghan and Ibrahim give an overview about notices of muqarnas found in the literature. Methods for a structural reading of muqarnas designs are not found. Decoding designs is based on heuristic methods; using experience and intuition leads to the proposal of interpretations.

1.2.3 **Outline of this Work**

In this work new methods are developed for doing research on muqarnas. We bring different disciplines, namely art history, mathematics and computer science, together for a better understanding of the muqarnas structure. For the first time a more mathematical context is used to define the muqarnas vaults and the muqarnas designs. This makes it possible to develop an algorithm that is able to analyze the muqarnas designs. The result is a reconstruction algorithm which is given as a software package. A faster understanding of a muqarnas design and a survey of the different reconstruction possibilities is realizable with this software. The reconstruction process is worked out for muqarnas vaults from the Seljuk and Il-Khanid time, fitting into domes or niches. Knowledge of art history concerning the common muqarnas styles corresponding to a certain region, time period or building materials is required to select the muqarnas reconstruction which fits best in the historical context.

The Il-Khanid muqarnas, as we know them from Iran, were often built from pre-fabricated elements. They mainly exist of a finite set of different elements. We will consider this property in the reconstructions. For the Seljuk muqarnas a wider variety of different elements is possible.

This work is organized as follows: besides of the introduction (Chapter 1), it contains three main parts, each represented by one chapter. The first part, which is presented in Chapter 2, is concerned with a formalization of the terms muqarnas vault and muqarnas design by defining these in a more mathematical context. Having clear definitions available makes it possible to analyze the muqarnas structure. This analysis is done in the second part, presented in Chapter 3. In that chapter we present the steps executed to calculate the muqarnas structure from its design. The reconstructing process is split in two steps: in the first step we construct a directed subgraph from the design of a muqarnas vault (see Section 3.1), and in the second step the muqarnas is reconstructed by reading the structure of this subgraph (see Section 3.3).

The third part (Chapter 4) focuses on the conversion of the reconstruction algorithm into a software tool. This tool is able to do suggestions for reconstructions of muqarnas designs. The program flows of the different parts the software consists of are explained. Additionally, a detailed description of the main data-structures and methods used in the software is given.

In Chapter 5, we show the results of applying the software tool for reconstructing different muqarnas vaults from the Seljuk and Il-Khanid periods. The last chapter (Chapter 6), contains a review in form of conclusions and proposals for further study. For studying the muqarnas structures we use terms from graph theory. The exact definitions can be found in the Appendix A.

Chapter 2

Structure of the Muqarnas

In this chapter we formalize the structure of muqarnas. In the first part (see Section 2.1), we study muqarnas as they appear in the Islamic architecture and formulate the properties we need to consider in our reconstructions. These properties are used in the second part (see Section 2.2) to define a muqarnas in a more mathematical way.

2.1 Muqarnas in the Architecture

In order to understand the structure of a muqarnas, we first introduce their building blocks, or muqarnas elements, as we find them in the Seljuk and Il-Khanid architecture (see Section 2.1.1). Our description of the elements is based on the definitions given by al-Kāshī. After presenting the elements, we give an overview about the way they can be combined for constructing a muqarnas vault (see Section 2.1.2).

2.1.1 Muqarnas Elements

In al-Kāshī's definition of muqarnas (see citation in Section 1.1.2), it is explained that a muqarnas is constructed from different cells. These cells are the main building blocks of which the muqarnas is built. Beside the cells, some space need to be filled with another kind of building blocks, the so-called intermediate elements. Al-Kāshī already

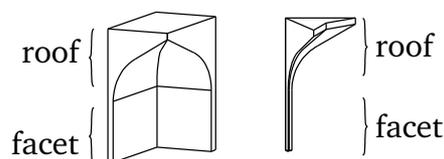


Figure 2.1. Examples of curved muqarnas elements. On the left: a cell. On the right: an intermediate element.

formulated a definition of a cell. In this definition a cell is compared with a house (see [Dold-Samplonius, 1992, p. 226]):

Definition 2.1.1 (cell, al-Kāshī). *The two (facets) can be thought of as standing on a plane parallel to the horizon. Both (facets) together with their roof are called one cell [bayt].*

We adopt from [Dold-Samplonius, 1992, p. 226] the word *cell* as translation for the Arabic word *bayt*. The word *bayt* could also be translated with *house*. Like a house, a cell is then split in its facets, which correspond to the straight part of the elements, and its roof, which correspond to the curved part (see Figure 2.1). In the context of al-Kāshī's manuscript such a division makes sense, because his aim was to calculate the surface of the muqarnas elements. Therefore, it was important to describe the exact curve on the side of the elements and calculate the surface of the roof and the facets separately.

We adopt the word *intermediate element* (Zwischenelement) from [Harb, 1978, p. 34] to describe the other type of building blocks. An intermediate element is an element which connects the roofs of two adjacent elements. These adjacent elements can be a cell or an intermediate element, as we see in Figure 2.2. The place between two elements is not necessarily filled with intermediate elements.

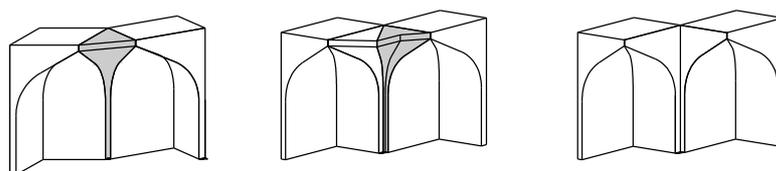


Figure 2.2. On the left: an intermediate element joining two cells. In the middle: an intermediate element joining a cell and an intermediate element. On the right: place between the cells is not filled with intermediate elements.

We often refer to the horizontal plane projection of an element. These projections are two dimensional polygons with three or four vertices, which represent the geometric figures we recognize if we watch a muqarnas from beneath (see Figure 2.3). We want



Figure 2.3. Portal to the enclosure at Sultan Han near Aksaray (Turkey). On the left: view from the front. In the middle: View from underneath. On the right: plane projection.

to emphasize that sides of elements become edges in the plane projection. Nodes in the plane projection are then the projection of lines where sides join (see Figure 2.4).

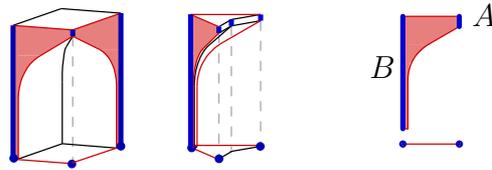


Figure 2.4. Elements with curved sides marked, together with their projections.

We often refer to the *curved sides* of the muqarnas elements. To describe the structure of the muqarnas the exact curve of the sides of the elements is not that important. But to know which sides are curved, gives us the orientation of the element and the place where the elements join. In our drawings we represent these sides indeed by curved faces as they are common in Il-Khanid muqarnas (see Figure 2.4). Without changing the structure of the muqarnas we could change their form (see e.g. [Yaghan, 2001a, p. 141]). To clear terminology, we define the *apex* of the curved side by the line at the front of the curved side. The *backside* of the curved side is the line on the opposite side. In the left picture of Figure 2.4, we marked the apex of the curved side with *A* and its backside with *B*. Both apex and backside of the curved side are in the plane projection represented by a node. Curved sides of a cell join in their apex, the curved sides of an intermediate element join in their backsides.

Muqarnas of the Il-Khanid period only consist of a small set of basic elements. These basic elements were already described by al-Kāshī (see [Dold-Samplonius, 1992]). In [Dold-Samplonius, 1996] an extensive explanation of his description is given. We give an overview of the plane projections of the basic elements of the Il-Khanid period in Figure 2.5. The names of the geometric figures are adopted from al-Kāshī. We see that the plane projection of the Il-Khanid muqarnas elements are figures based on a *square* (left column) or on a *rhombus* (right column). We choose our unit according to the edges of the square. The length of this side will be our measure unit and we adopt from al-Kāshī the word *module* [*miqyās*] for its size (see [Dold-Samplonius, 1992, p. 226]).

Concerning square based elements, we find a *square* with all sides equal to the module. Elements with a square as plane projection can appear as a cell or as an intermediate element. Furthermore, we find a *jug* with smaller diagonal as the square. In Il-Khanid muqarnas a jug only appears as plane projection of a cell. The longer sides of the jug have the module as their length, these are the sides which correspond to the curved sides of the element. The length of the smaller diagonal of a jug equals the module. The *large biped* is what remains if we subtract the jug from the square. In Il-Khanid muqarnas we only find a large biped as plane projection of an intermediate element, in general in connection with a jug. The remaining square-based figure is the *half square*, where the square is cut in its diagonal. Elements with a half square as plane projection in general appear in the first tier as a cell., but may also appear as an intermediate element.

Based on a rhombus, we first consider the *rhombus* itself. The rhombus has four sides with length equal to the module, two opposite angles of 45° and two opposite

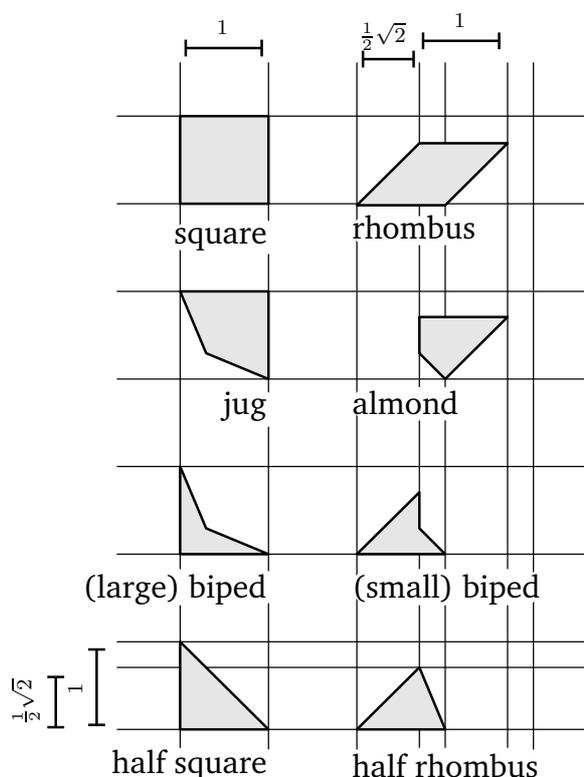


Figure 2.5. Overview of the basic muqarnas elements.

angles of 135° . The rhombus appears as plane projection of a cell and of an intermediate element. In Figure 2.6 three different elements with a rhombus as plane projection are visualized. In the first picture we see a cell with plane projection a rhombus, the curved sides join in the apex with an angle of 135° . In the right two pictures of Figure 2.6 we see two different intermediate elements. In the first the shorter diagonal is used for the orientation of the element and the curved sides join in an angle of 135° . In the second picture the larger diagonal is used, and the curved sides join in an angle of 45° . In Il-Khanid muqarnas we do not find cells with a rhombus as plane projection where the larger diagonal is used for the orientation. The *almond* and the *small biped* represent a



Figure 2.6. Three different elements with a rhombus as plane projection.

possibility to split the rhombus, like a jug with a large biped for the square. In Il-Khanid muqarnas the almond appears only as plane projection of a cell. This cell has two sides

with sizes equal to the module, which are the sides corresponding to its curved sides. These two sides join in an angle of 45° . The opposite angle is 135° and the other two angles have both sizes of 90° . A small biped is what remains if we subtract an almond from a rhombus. The two sides with sizes equal to the module correspond to the curved sides of the element. These curved sides join in an angle of 45° and the opposite angle equals 225° . The remaining two angles have sizes of 45° . In Il-Khanid muqarnas, a small biped only appears as plane projection of an intermediate element. The last figure based on a rhombus is the half-rhombus. The half-rhombus is given by dividing the rhombus over its smaller diagonal yielding an equilateral triangle. In Il-Khanid muqarnas, it only appears as plane projection of an intermediate element. Often we find two half-rhombi in combination with a jug and a square forming a hexagon (see Figure 2.7).

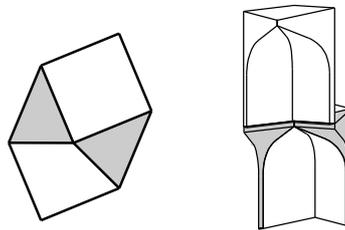


Figure 2.7. Common combination where half-rhombi are used to carry a jug.

In Il-Khanid muqarnas other elements than the basic ones only appear at the boundary. In Figure 2.8 the plane projection of an Il-Khanid muqarnas vault in a niche of the Friday Mosque in Ashtarjān is drawn. We see that this projection indeed mostly consists of polygons as given in the overview in Figure 2.5. At the part corresponding to the front of the muqarnas vault other muqarnas elements appear. They need to be adapted so that the muqarnas fits into the vault. In the center of the upper most tier we often find ele-

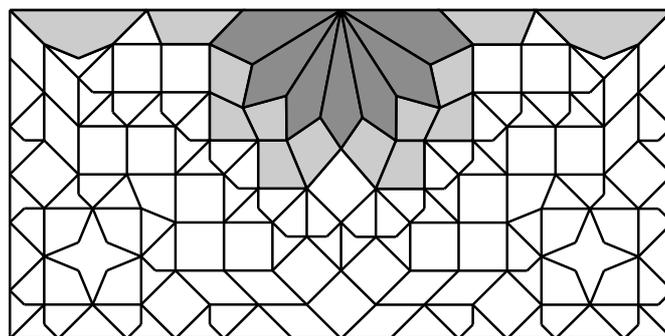


Figure 2.8. Plane projection of a muqarnas vault in the entrance portal of the Friday Mosque (1315–1316) in Ashtarjān (Iran). We find figures corresponding to non-basic elements (gray) at the front of the vault. The center is filled with barley-kernels (dark gray).

ments with *barley kernels* as plane projection. A barley kernel has two sides with length of the module. The length of the diameter varies and needs to be adapted to the vault in which the muqarnas is built. In Il-Khanid muqarnas, it only appears as plane projection of intermediate elements of the upper tier.

It is a special property of Il-Khanid muqarnas that they are formed from only a few different kind of elements. In muqarnas of other styles we find more variation in the elements. For example in the Seljuk muqarnas we see that there is a larger freedom for varying the angles (see e.g. Figure 2.9). In the designs found in the Topkapı Scroll (see [Necipoğlu, 1995]) this is also the case, but the structure of the muqarnas remains the same.

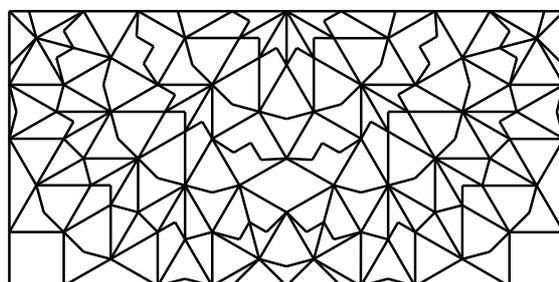


Figure 2.9. Plane projection of the muqarnas in the entrance portal of the Hunat Hatun Camii (1238) in Kayseri (Turkey).

In other kinds of muqarnas, for example the ones in Morocco, besides of the cells and intermediate elements other types of elements appear. In [Castéra, 1996, p.316–319] an overview of the elements common in muqarnas used in Morocco is given. As example, we mention the rectangular element as given in Figure 2.10. We see that this is neither a cell nor an intermediate element. In this element the curved sides do not join, but they lie opposite to each other. As we did not find such elements in Il-Khanid and Seljuk muqarnas it will not be considered in our reconstructions.

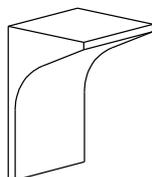


Figure 2.10. Rectangular element as used in muqarnas in Morocco.

2.1.2 Three-Dimensional Muqarnas Structure

In this section we study the structure of muqarnas by discussing how the different muqarnas elements of a muqarnas can be arranged to form a muqarnas structure. In al-Kāshī's

definition, a muqarnas is compared with a staircase. We have to think about a structure constructed from building blocks arranged in different levels, which are the stairs. Al-Kāshī calls such a level a *tier* [*ṭabaqqa*] and we adopt this terminology. He defines a tier as follows (see [Dold-Samplonius, 1992, p.226]):

Definition 2.1.2 (tier). *Adjacent cells, which have their bases on one and the same surface parallel to the horizon are called one tier.*

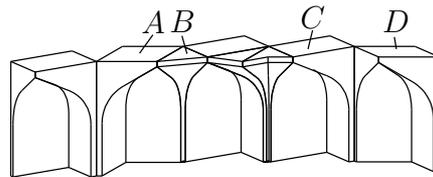


Figure 2.11. Part of a tier of the muqarnas niche in the basement of the north iwan in the Friday Mosque of Natanz.

In Figure 2.11 we see a part of a tier. We observe that the elements on the same tier join together at their curved sides. The curved sides can join directly together, like we see in the case of the elements *A* and *B*, or they can join only at the backside of the curved side, so that there remains an angle between the curved sides of two adjacent elements (elements *C* and *D*). Cells can *only* join other elements in the same tier at their curved sides. Intermediate elements most often join other elements on the same tier at their curved sides, but also their front parts can join front parts of intermediate elements at the same tier (see Figure 2.12). In that case the curved sides still join elements on the same tier, and the front sides “touch” each other. The two front parts of a full intermediate

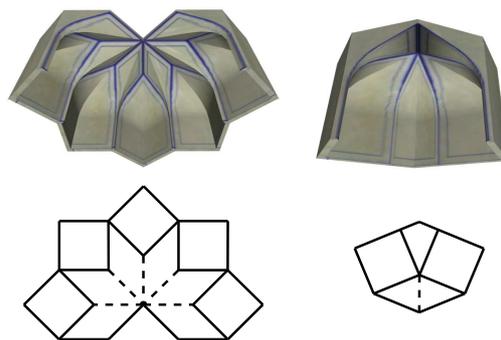
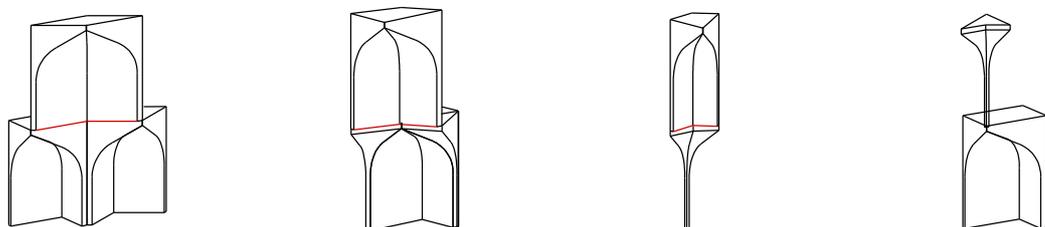


Figure 2.12. Examples of structures where front sides of intermediate elements touch other intermediate elements in the same tier.

elements touch two different elements. The situation as shown in Figure 2.12, where two intermediate elements lie opposite of each other and touch in their front, only appears for half intermediate elements, and not for full intermediate elements.

The next tier will then be arranged on the previous one in such a way that in the projection no gaps will be left and the projection of the elements do not overlap. In Figure 2.13 we demonstrate how cells are built upon elements a tier below. We find that the apex of a curve always ends in the bottom of a curved side of an element in the next tier. Because of this property we say that a muqarnas is oriented in upwards direction: following the curved sides gives a possibility to go up in the muqarnas. Further, we see in Figure 2.13(a) that the backsides of a cell can be put on curved sides of elements below. The Figures 2.13(b) and 2.13(c) show that backsides of a cell also can be put on front sides of intermediate elements. An intermediate element stands at the front of a cell from the previous tier.



(a) Back on curve.

(b) Back on front.

(c) Back on front.

(d) Back on front.

Figure 2.13. Different ways of building cells upon elements in a lower tier.

We summarize the possibilities how we can combine the different elements:

curve to curve: A curved side of an element joins a curved side of an element on the same tier (see Figure 2.11).

front to front: The front part of an intermediate element can touch the front part of an intermediate element on the same tier (see Figure 2.12).

back on curve: The backside of a cell can stand on the curved side of an element of the tier below (see Figure 2.13(a)). The apex of the curved side of the element in the lower tier then joins the curved sides of the cell in the upper tier (curve points to curve).

back on front: The backside of a cell can stand on the front of an intermediate element of the tier below (see Figures 2.13(b) and 2.13(c)). An intermediate element stands on the front of a cell (see Figure 2.13(d)).

Considering the complete muqarnas structure, we define the *bottom* of the muqarnas as the set of elements which do not stand on elements of a lower tier. This bottom is called *regular* if all elements of the bottom are contained in the first tier of the muqarnas.

Similar, the *center* of the muqarnas consists of the set of elements which do not join to elements in a upper tier. The center is called *regular* if all elements of the center belong to the most upper tier of the muqarnas. The *front* part of a muqarnas fitting into a niche is formed by the elements at the border of the tiers.

2.2 Muqarnas as an Abstract Geometrical Structure

To define a muqarnas in a more mathematical context, we define it as a finite set of muqarnas elements, together with their position and orientation. We will start this section with a more abstract definition of muqarnas elements, and try to formulate some properties for the “set of muqarnas elements” to make clear what kind of sets are allowed (see Section 2.2.1). In this more abstract characterization of muqarnas only the structure of the muqarnas is considered, no information about the exact appearance (material, color, measurements) of the muqarnas is given. We continue with an definition of the muqarnas plan, which will substitute the muqarnas design (see Section 2.2.2). In the last part of this chapter (see Section 2.2.3), we calculate the proportions of the elements appearing in Il-Khanid muqarnas and formulate special properties of an Il-Khanid muqarnas plan.

2.2.1 Elements as Abstract Objects

The crucial sides of the elements for describing the structure of the muqarnas are the curved ones. In the previous section we have seen that these are the surfaces where elements on the same tier are connected to each other. We see in Figure 2.1 that the two curved sides of a cell join at the front of the element, in the apex of the curved sides. The two curved sides of an intermediate element join at the backsides of the curved sides. In Figure 2.4 we see that the plane projection of the place where the curved sides join is a node and we call this node the *central node* of the element.

Definition 2.2.1 (central node). *The central node of an element is the plane projection of the place where the curved sides join.*

To classify the elements by means of parameters such as an angle and some typical length, we use the *angle* between the curved sides of the element and the *diameter*. In the case of a full element the diameter is defined as being the distance between the central node and its opposite node. In the case of a half element, where the projection only consists of three vertices, there is no diameter. Before formalizing these definitions, we introduce the notation $\alpha(c, v, w)$ for the size of the angle in radians over which we can rotate v in counter-clockwise direction over c , so that vector $v - c$ and vector $w - c$ join (see Figure 2.14).

Let $c \in \mathbb{R}^2$ be the central node of an element and $v = (v_x, v_y), w = (w_x, w_y) \in \mathbb{R}^2$ such that the edge $\text{edge}(c, v)$ connecting c and v and the edge $\text{edge}(c, w)$ connecting c and w



Figure 2.14. To join $v-c$ with $w-c$ we have to rotate v over c by an angle of $\alpha(c, v, w)$ in counter-clockwise direction.

correspond to the curved sides of a muqarnas element. The nodes v and w are chosen in the order so that if we rotate v in counter-clockwise direction to w we cross over the projection of the interior of the element.

Definition 2.2.2 (standard position). We say that the element is in standard position if its central node c lies in the origin and $\text{edge}(c, v)$ joins with the x -axis, so $c = (0, 0)$ and $v_y = 0$.

The left two pictures of Figure 2.15 show the plane projection of muqarnas elements in standard position. The curved sides are represented by arrows. This motivates to classify the type of a basic muqarnas element as follows:

Definition 2.2.3 (type). The type of a muqarnas element is given by two nodes $a = (a_x, a_y), b = (b_x, b_y) \in \mathbb{R}^2$, representing the curved sides and, in case of a full element, a number $d \in \mathbb{R}_{\geq 0}$, representing the diameter, so that $a_y = 0$ and $0 < \alpha(0, a, b)$.

The type as formulated in this definition is then given by the element in standard position, it is given by the two edges $\text{edge}(0, a)$ and $\text{edge}(0, b)$ joining in the central node. The size of the diameter of the element equals d . In case $a_x < 0$ we are dealing with a cell and if $a_x > 0$ the figure corresponds to an intermediate element. The fourth node of the element can be found by drawing a line from 0 with end-node k so that the distance $\|k\|$ from k to 0 is equal to d and $\alpha(0, a, k) = \frac{1}{2}\alpha(0, a, b)$. If we are dealing with a half element, there is no diameter.

The right two pictures of Figure 2.15 illustrate the definition of the position of an element. We calculate c and ϕ from the plane projection. The height $h \in \mathbb{Z}_{\geq 1}$ of the element is the number of the tier to which the element belongs. The tiers are counted from below, so that the elements in the bottom tier have height $h = 1$. The position $c \in \mathbb{R}^2$ will be the position in the plane projection, given by the coordinates (c_x, c_y) of the central node. The orientation of the element is given by $\phi \in [0, 2\pi)$. Let $c, v, w \in \mathbb{R}^2$ as before, so that the edges $\text{edge}(c, v)$ and $\text{edge}(c, w)$ are the projections of the curved sides of the muqarnas element and the element lies at the side we cross over by rotating v counter-clockwise over c to w . We define $e_0 := (1, 0)$. In case of a cell the position of the element is given by $\{c, h, \phi\}$ with $\phi = \alpha(c, c - e_0, v)$, so the element is rotated from standard position in counter-clockwise direction over ϕ and translated over c . Similar, the position of an intermediate element is given by $\{c, h, \phi\}$ with $\phi = \alpha(c, c + e_0, v)$.

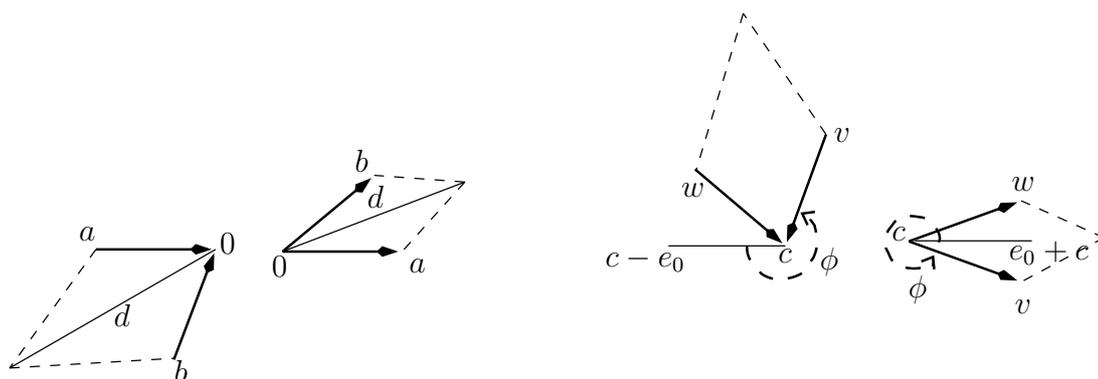


Figure 2.15. On the left: plane projection of a cell and an intermediate element in standard position. On the right: the projection of a cell and an intermediate element with position $\{c, h, \phi\}$.

Definition 2.2.4 (position). Consider an element in tier h with central node c . Let $\text{edge}(c, v)$ be the projection of a curved side of the element, such that if we rotate $\text{edge}(c, v)$ in the plane in counter-clockwise direction, then we cross over the projection of the element. The position of the element is then given by $\{c, h, \phi\}$ with $\phi = \alpha(c, c - e_0, v)$ for a cell and $\phi = \alpha(c, c + e_0, v)$ for an intermediate element.

2.2.2 Muqarnas Plan

We have seen in Section 1.1.3 that a muqarnas design is a drawing which consists of polygons representing the plane projections of the muqarnas elements. A muqarnas design therefore mainly consists of geometrical figures as given in Figure 2.5: the plane projections of the building blocks of the muqarnas. In practice a muqarnas design does not correspond to the exact projection of all the elements. If we study the plate found at the Takht-i-Sulaymān, we observe that only square and rhombus figures are drawn. During the analysis of this design we find that it is not possible to construct a muqarnas only consisting of squares and rhombi; we have to interpret some rhombi as a combination of an almond and a biped (see e.g. [Dold-Samplonius and Harmsen, 2005]). In the muqarnas designs of the Topkapı Scroll the designers mark where figures have to be split. In these designs, polygons often appear with more than four nodes, which suggests that the projection of the sides of some elements is not always drawn. We see, for example, in Figure 2.16 a detail from the left lower corner of the design nr. 16 from the Topkapı Scroll (see [Necipoğlu, 1995, p. 294]). In the right picture a dotted line is inserted to mark a side of an element which is not drawn in the design, but necessary for the construction.

In the software we developed, we use a muqarnas plan as an input. A muqarnas plan is the set of lines corresponding to the plane projection of all sides of the elements. The definition of muqarnas plan is inspired by the muqarnas designs we know, but we need to be aware that we often have to adapt the design before it is a muqarnas plan containing

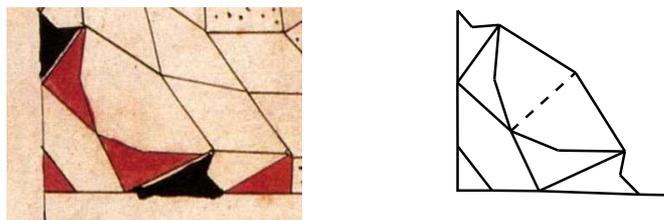


Figure 2.16. Part of a design from the Topkapı Scroll (from [Necipoğlu, 1995, p. 294]).

the plane projection of all sides of the elements. We define a muqarnas plan as a graph (see Definition A.1). Interpreting it as a graph gives us a way to store the structural information contained in the plan. As a graph the edges E and nodes of intersection N are stored.

Definition 2.2.5 (muqarnas plan). A muqarnas plan $P = P(N, E)$ is a two dimensional drawing, existing of a finite set of edges E and a finite set of nodes N such that if $l, m \in E$, then $l \cap m \in N$.

To understand the meaning of the edges in the muqarnas plan we first remark that the edges in the projection of a cell correspond to the two curved sides and the two backsides of the cell. The edges in the projection of an intermediate element correspond to the projection of the two curved sides and the two front parts of the intermediate element. In case of a half element, only one edge in the projection corresponds to the backside respectively the front part of the element. An edge in the interior of a plan corresponds to a place where two elements of the muqarnas join. We already listed all cases in which sides of elements join to sides of other elements in Section 2.1.2. Each edge in the interior of the muqarnas plan then corresponds with one of the situation as listed there; curve to curve, front to front, back on curve or back on front.

The geometrical information is included in the information of the coordinates of the nodes N . The structure information is presented by the connection between these nodes, which is given by the set of edges E . In this way the plane projections of the elements are not explicitly stored in the plan, but can be calculated from the coordinates of the nodes and their connections. We call the plane projection of a muqarnas element a *figure*. As each muqarnas element has three or four nodes in its plane projection, the figures are the cycles in P of length three or four. They will be given by ordered sets $\{u, v, w\}$ such that if we rotate edge (v, w) over v to edge (v, u) in counter-clockwise direction, we cross over the figure defined by $\{v, w, u\}$. In case of a figure consisting of three nodes this means that edge (w, u) must exist and $\alpha(v, w, u) < \pi$. In case of a figure corresponding to a polygon of four nodes there exists a node k connected to w and u with $\alpha(v, w, k) < \alpha(v, w, u)$ as we need to cross the polygon given by v, w, k and u during rotating w to u . We say that the figure is oriented in counter-clockwise direction as a walk over u, v, w, k, u defines a counter-clockwise cycle (see Figure 2.17).

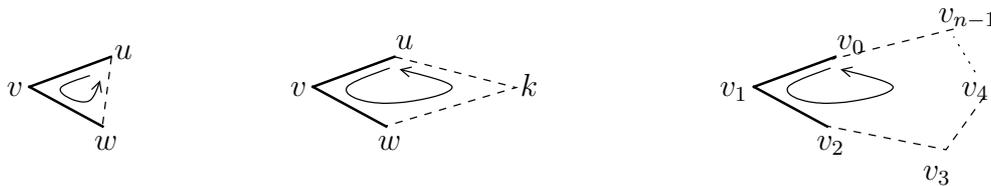


Figure 2.17. On the left: two examples of ordered triples $\{u, v, w\}$ defining a figure. On the right: example of an ordered triple $\{u, v, w\}$ defining an island.

Definition 2.2.6 (figure). *The ordered triple $\{u, v, w\}$ defines a figure if nodes v and w are connected and $\alpha(v, w, u) < \pi$ or if nodes v and w have a common adjacent $k \neq v$ and $\alpha(v, w, k) < \alpha(v, w, u)$.*

In the next chapter, we also work with a subgraph of the plan P by removing certain edges from the plan. Thereby polygons arise which consists of more than four nodes and do not fit in our definition of a figure. We call these polygons *islands* (see right picture of Figure 2.17). In this context an island is a generalization of a figure and we also define them by ordered triples $\{u, v, w\}$. The corresponding polygon is then given by a sequence $\{v_0, \dots, v_n\}$ starting with u, v, w , so $v_0 = u, v_1 = v, v_2 = w$. For being an island this sequence needs to define a counter-clockwise oriented cycle in the plan, such that the polygon given by this cycle does not contain other nodes of the plan P . Because it defines a cycle $v_n = v_0$ and being counter-clockwise oriented means that by rotating v_{i+1} over v_i to v_{i-1} for $i = 1, \dots, n-1$ in counter-clockwise direction we cross over the polygon given by v_0, \dots, v_n .

Definition 2.2.7 (island). *An ordered set $\{u, v, w\}$ defines an island in $P(N, E)$ if there exist a counter-clockwise oriented sequence $v_0, \dots, v_n \in N$ with $v_0 = u, v_1 = v, v_2 = w$ and $v_n = v_0 = u$ such that*

$$\left\{ t_0 v_0 + \dots + t_n v_n : \sum_{i=0}^n t_i = 1, 0 \leq t_i < 1 \text{ for } i = 1, \dots, n \right\} \cap N = \emptyset. \quad (2.1)$$

We need the condition (2.1) so that polygons with nodes in the interior are excluded from being an island.

We use in general a quarter muqarnas plan for our input. With a quarter plan we mean the part of which four copies are needed, so that a dome can be filled with the muqarnas (see Figure 2.18). We get these copies by mirroring the quarter plan twice. If we mirror it once, we get a plane projection of a muqarnas corresponding to a niche. In that case a quarter plan is the projection of half the muqarnas. We always choose the orientation of a quarter muqarnas plan such that the left lower corner of the plane projection corresponds to the right corner at the back of the muqarnas if we stand in front of a muqarnas niche (see Figure 2.19). The diagonal nodes of a quarter muqarnas plan are then the nodes on the line from the left lower corner to the right upper corner of

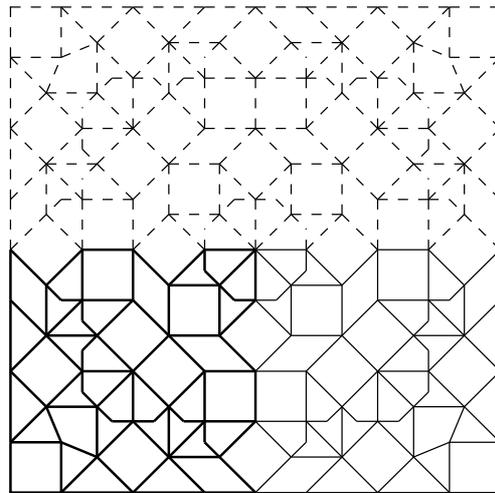


Figure 2.18. The thick lines represent a quarter of a muqarnas plan. The complete plan could be the projection of a muqarnas vault of the south octagon at the Takht-i-Sulaymān (see [Harb, 1978, p. 46]).

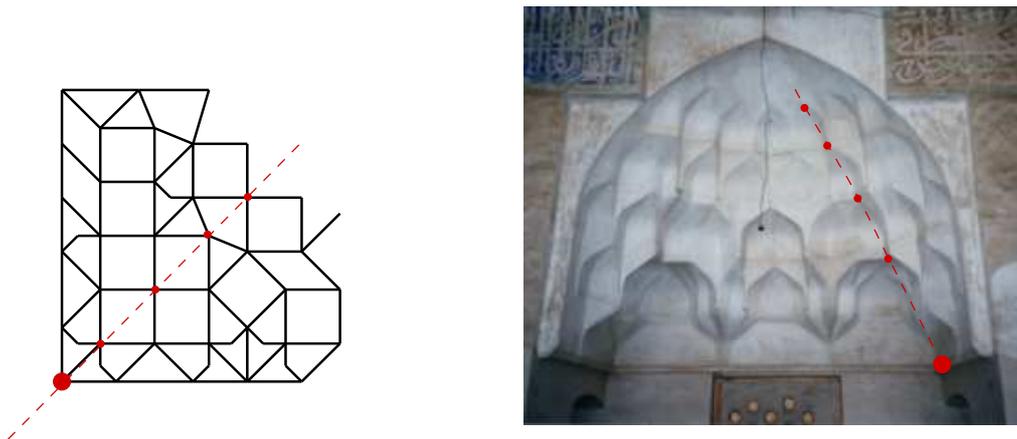


Figure 2.19. Plan and photo of muqarnas in the basement of the north iwan from the Friday Mosque in Natanz. In red the diagonal nodes.

the plan. If the muqarnas plan corresponds to a muqarnas vault in a dome, this diagonal gives a symmetry line of the plan. For niches this is not the case.

In the next chapter we will see that for converting a muqarnas plan into a muqarnas structure, it seems to be important to know how a certain node in a muqarnas plan is situated according to the boundary of the plan. The definitions concerning the boundary are visualized in Figure 2.20. The boundary of a plan consists of the nodes which are not surrounded by figures or islands. This means that v is in the boundary if it has neighbors u, w such that $\{u, v, w\}$ and $\{w, v, u\}$ do not define a figure or island. We denote the set of boundary nodes of a plan P by $W(P)$.

Definition 2.2.8 (boundary). *The boundary $W(P) \subset N$ of a muqarnas plan $P(N, E)$ is the set of nodes $n \in N$ which are not surrounded by figures or islands*

In this way the boundary is the collection of nodes not belonging to the inner part of the plan. The boundary of a muqarnas plan as defined here, consists of nodes corresponding to vertices at the bottom, the side, or the center of the muqarnas. We therefore split the boundary of a plan also in the center and the bottom boundary.

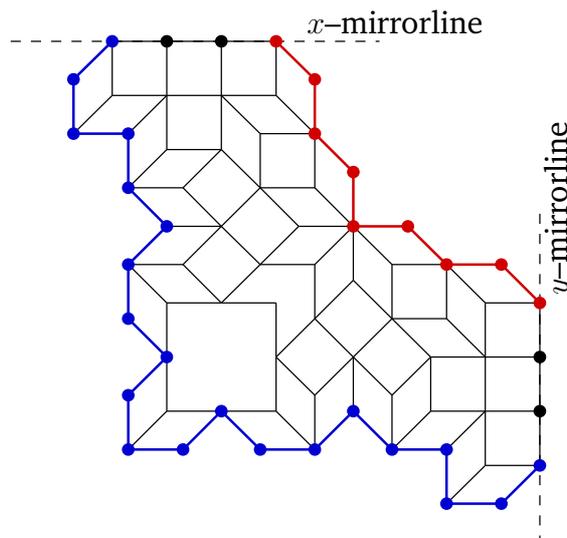


Figure 2.20. The bold nodes represent the boundary of the plan corresponding to the vault over the east portal of the shrine in Bistam. The blue part defines the bottom boundary and the red part the center.

To separate the different parts of the boundary we assume that the plan is a quarter plan with the usual orientation. We consider the plan in the xy -plane and represent its nodes with (x, y) coordinates. Let $x_{\max} \in \mathbb{R}$ be the maximal x -value appearing in the plan and $y_{\max} \in \mathbb{R}$ be the maximal y -value. We define the x -mirrorline parallel to the x -axis by the line $y = y_{\max}$ and the y -mirrorline parallel to the y -axis by $x = x_{\max}$.

The boundary of the x -mirrorline is given by the nodes on the x -mirrorline with maximal or minimal x -value. The interior of the x -mirrorline is what remains after removing the boundary of the x -mirrorline. The boundary and interior of the y -mirrorline is defined analogously.

Definition 2.2.9 (center). *The center $W_c(P)$ of the plan P is the shortest path in $W(P)$ between the end-node of the x -mirrorline with maximal x -value and the end-node of the y -mirrorline with maximal y -value.*

Definition 2.2.10 (bottom boundary). *The bottom boundary $W_{bb}(P)$ of the plan P is given by the path in $W(P)$, not containing center nodes, between the end-node of the x -mirrorline with minimal x -value and the end-node of the y -mirrorline with minimal y -value.*

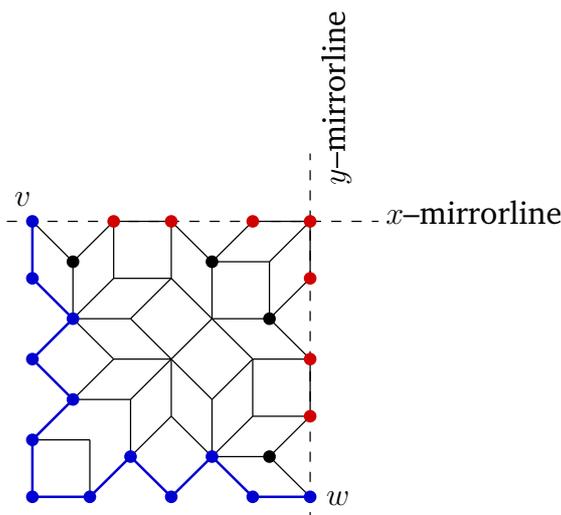


Figure 2.21. The bottom boundary nodes are given by the blue nodes in this plan. They are found by removing the red nodes and then the path between v and w is calculated over the remaining boundary nodes.

In Figure 2.21 we see the motivation to define the bottom boundary in this way. It is defined by the path between the node v on the the x -mirrorline with smallest x -value and the node w on the y -mirrorline with smallest y -value. By excluding the nodes (marked red in Figure 2.21) on the center and the mirrorlines, we find the path at the right side.

We will see that often there are several reconstructions possible from a given plan. These reconstructions differ in their global form. The shape can often be fixed by restrictions on the boundary. We say that a given muqarnas has a *regular bottom boundary* if and only if all nodes in the bottom boundary of the plan correspond to elements in the first two tiers of the muqarnas. Similarly we define the center of a muqarnas as being *regular* if and only if all nodes in the center of the muqarnas plan correspond to elements in the top tier.

2.2.3 Structure of Il-Khanid Muqarnas

We finish this chapter by applying our definitions of the muqarnas elements and the muqarnas plan to Il-Khanid muqarnas. From the plane projections (see Figure 2.5) of the basic muqarnas elements we can calculate their angles and diameters. In Table 2.1 an overview of these angles and diameters is given by using the module as our unit of measurement.

Il-Khanid muqarnas are designed by mainly using the basic elements. This has as consequence that the coordinates of the nodes in their plane projection can be represented by integers. In Chapter 4 we will see how we can use this integer representation in our software as a data-structure to represent a muqarnas plan. An advantage of this integer representation is that it is possible to verify whether nodes are the same by comparing

name	angle	diameter
square	$\frac{1}{2}\pi$	$\sqrt{2}$
jug	$\frac{1}{2}\pi$	1
large biped	$\frac{1}{2}\pi$	$\sqrt{2} - 1$
half square	$\frac{1}{2}\pi$	–
(long) rhombus	$\frac{1}{4}\pi$	$\frac{1}{\sqrt{2} \sin(\pi/8)}$
(short) rhombus	$\frac{3}{4}\pi$	$\frac{1}{\sqrt{2} \sin(3\pi/8)}$
almond	$\frac{1}{4}\pi$	$\frac{1}{\cos(\pi/8)}$
small biped	$\frac{1}{4}\pi$	$\frac{1}{\sqrt{2} \cos(\pi/8)}$
half rhombus	$\frac{1}{4}\pi$	–

Table 2.1. Overview of the measures of the basic muqarnas elements.

integers instead of floating–point numbers. Because floating–point numbers are rounded to integers, we can work with drawings which are not very exact: the nodes are rounded to the nodes fitting in a muqarnas plan.

We can verify the following properties for the basic elements by checking these statements for all nodes in the projections drawn in Figure 2.5:

- The plane projection of the elements are polygons of which all angles have sizes $k\pi/8, k \in \mathbb{Z}$.
- Nodes of the plane projections of the basic elements in standard position can be represented by coordinates in the grid

$$\mathbb{L} := \left(\mathbb{Z} + \mathbb{Z}/\sqrt{2} \right) \times \left(\mathbb{Z} + \mathbb{Z}/\sqrt{2} \right).$$

We prove that nodes of a plan consisting only of basic elements are all in the grid \mathbb{L} .

Proposition 2.2.11. *The set of nodes N of a plan $P(N, E)$ corresponding to a muqarnas only consisting of basic elements, is a subset of the grid $\mathbb{L} := (\mathbb{Z} + \mathbb{Z}/\sqrt{2}) \times (\mathbb{Z} + \mathbb{Z}/\sqrt{2})$.*

PROOF. We give the proof by showing that we only need two transformations for putting the polygons from their standard position to their places in the muqarnas plan. These two standard transformations are given by:

- (i). Translation over $n \in \mathbb{L}$.
- (ii). Rotation by $k\pi/4, k \in \mathbb{Z}$.

We first prove that these two transformations are homomorphisms on \mathbb{L} if we consider \mathbb{L} as a \mathbb{Z} -module. Then we show that these transformations are sufficient to bring the polygons to their position in the plan.

Transformation (i) is an operation in \mathbb{L} : let $l, n \in \mathbb{L}$, we can translate l over n by adding n to l . It can be confirmed that $n + l \in \mathbb{L}$. For transformation (ii), let $l \in \mathbb{L}$ and let R_k be the rotation-matrix for rotating over an angle of $k\pi/4$, $0 \leq k \leq 8$, hence

$$R_k = \begin{pmatrix} \cos(k\pi/4) & -\sin(k\pi/4) \\ \sin(k\pi/4) & \cos(k\pi/4) \end{pmatrix} = \begin{pmatrix} \pm 1/\sqrt{2} & \pm 1/\sqrt{2} \\ \pm 1/\sqrt{2} & \pm 1/\sqrt{2} \end{pmatrix}.$$

Because all entries of R_k are equal to $\pm 1/\sqrt{2} \in \mathbb{Z} \times \mathbb{Z}/2$, we can confirm that $R_k \cdot l \in \mathbb{L}$.

To prove that these are the only transformations we need, we use induction on the number of elements in the muqarnas. If a muqarnas consists of one element, we choose the orientation of the grid such that this element is in standard position. For a plan of $n + 1$ elements, we look at a polygon at the boundary of the plan. The element corresponding to this polygon needs to join with its curved side to another element. Therefore, one of the edges of the polygon has length 1. Let this edge be $\text{edge}(v, w)$. As the other n polygons have coordinates in the grid \mathbb{L} , the end-nodes v, w of the edge $\text{edge}(v, w)$ are in the grid \mathbb{L} . One of the end-nodes of $\text{edge}(v, w)$ corresponds to the central node. We assume that this is the node v . By translating over $-v$, the central node lies in the origin. The coordinates of the other end-node $w - v$ is then given by $w - v = (a + b/\sqrt{2}, c + d/\sqrt{2})$ with $a, b, c, d \in \mathbb{Z}$. As the length of the edge $\text{edge}(v, w)$ equals to 1 we find that

$$\left\| \left(a + b/\sqrt{2}, c + d/\sqrt{2} \right) \right\| = 1,$$

which gives us that

$$a^2 + \frac{1}{2}b^2 + c^2 + \frac{1}{2}d^2 + (ab + cd)\sqrt{2} = 1,$$

so we find that

$$a^2 + \frac{1}{2}b^2 + c^2 + \frac{1}{2}d^2 = 1 \wedge ab + cd = 0.$$

The only integer solutions are now $(a, b, c, d) = (\pm 1, 0, 0, 0)$, $(a, b, c, d) = (0, 0, \pm 1, 0)$ and $(a, b, c, d) = (0, \pm 1, 0, \pm 1)$. This means that

$$\left(a + b/\sqrt{2}, c + d/\sqrt{2} \right) = (\pm 1, 0),$$

or

$$\left(a + b/\sqrt{2}, c + d/\sqrt{2} \right) = (0, \pm 1),$$

or

$$\left(a + b/\sqrt{2}, c + d/\sqrt{2} \right) = (\pm 1/\sqrt{2}, \pm 1/\sqrt{2}).$$

The node $w - v$ is therefore given by rotating $(1, 0)$ or $(-1, 0)$ over the origin by $k\pi/4$. Let the other edge of length one corresponding with a curved side be given by $\text{edge}(v, u)$,

then the position of the polygon p is given by translating with v and rotating over $\frac{k}{4}\pi$ or over $k\pi/4 \pm \alpha(v, u, w)$, As $\alpha(v, u, w)$ is the angle of the element, $\alpha(v, u, w) = k\pi/4$ for a $k \in \mathbb{Z}$. \square

We use the property that the grid $\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ is countable to represent each node in \mathbb{L} with one integer. We define the function $f : \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ by

$$f(i, j) \mapsto j + \frac{1}{2}(i + j)(i + j + 1). \quad (2.2)$$

we denote $\lfloor r \rfloor$ for the largest integer smaller than or equal to r .

Proposition 2.2.12. *The function f is invertible with the inverse $g : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ given by*

$$g : k \mapsto \left(n - k + \frac{1}{2}n(n + 1), k - \frac{1}{2}n(n + 1) \right) \quad (2.3)$$

with $n = \lfloor -\frac{1}{2} + \frac{1}{2}\sqrt{8k + 1} \rfloor$.

PROOF. We define g as in (2.3) and have $g(\mathbb{Z}_{\geq 0}) \subset \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ and $f(\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}) \subset \mathbb{Z}_{\geq 0}$. It suffices to prove that $g(f(i, j)) = (i, j)$ for all $(i, j) \in \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ and $g(f(k)) = k$ for all $k \in \mathbb{Z}_{\geq 0}$.

Let $(i, j) \in \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$, then

$$\begin{aligned} g(f(i, j)) &= g\left(j + \frac{1}{2}(i + j)(i + j + 1)\right) \\ &= \left(n - k + \frac{1}{2}n(n + 1), k - \frac{1}{2}n(n + 1) \right), \end{aligned} \quad (2.4)$$

with $k = j + \frac{1}{2}(i + j)(i + j + 1)$ and with $n = \lfloor -\frac{1}{2} + \frac{1}{2}\sqrt{8(j + \frac{1}{2}(i + j)(i + j + 1)) + 1} \rfloor$.

We note that

$$\frac{1}{2}\sqrt{8(j + \frac{1}{2}(i + j)(i + j + 1)) + 1} = \sqrt{(i + j + 1)^2 - i + j - \frac{3}{4}}$$

and

$$i + j + \frac{1}{2} \leq \sqrt{(i + j + 1)^2 - i + j - \frac{3}{4}} < i + j + \frac{3}{2},$$

which can be seen by writing $(i + j + 1)^2$ as $((i + j + \frac{1}{2}) + \frac{1}{2})^2 = (i + j + \frac{1}{2})^2 + (i + j + \frac{1}{2}) + \frac{1}{4}$ for the left estimate and $(i + j + 1)^2$ as $((i + j + \frac{3}{2}) - \frac{1}{2})^2 = (i + j + \frac{3}{2})^2 - (i + j + \frac{3}{2}) + \frac{1}{4}$ for the right estimate. We find

$$n = \lfloor -\frac{1}{2} + \frac{1}{2}\sqrt{8(j + \frac{1}{2}(i + j)(i + j + 1)) + 1} \rfloor = i + j.$$

By setting $n = i + j$ in (2.4) we can verify that $g(f(i, j)) = (i, j)$. To prove that $f(g(k)) = k$ we calculate $f(i, j)$ with $i = n - k + \frac{1}{2}n(n + 1)$ and $j = k - \frac{1}{2}n(n + 1)$. We can verify that $i + j = n$ and therefore $f(i, j) = k - \frac{1}{2}n(n + 1) + \frac{1}{2}n(n + 1) = k$. \square

The function f is only invertible on $\mathbb{Z}_{\geq 0}$ and not on \mathbb{Z} , we translate the muqarnas plan so that all nodes can be given by non-negative integers $a, b, c, d \in \mathbb{Z}_{\geq 0}$. We first translate the plan so that the smallest x -value and the smallest y -value both equals 0. Then we can represent the coordinates $v = (a + b/\sqrt{2}, c + d/\sqrt{2})$ by $(a, b, c, d) \in \mathbb{Z}_{\geq -1} \times \mathbb{Z}_{\geq -1} \times \mathbb{Z}_{\geq -1} \times \mathbb{Z}_{\geq -1}$. We need the “-1” to be able to represent all coordinates of the biped. In standard position the node opposite of the central node of the small biped is given by $v = (1/\sqrt{2}, 1 - 1/\sqrt{2})$, so we need $d = -1$ to represent its coordinates. The node opposite of the central node of the large biped in standard position is given by $v = (1 - 1/\sqrt{2}, 1 - 1/\sqrt{2})$ and so $b = d = -1$. To be sure that we do not need to use negative integers, we add 1 to all a, b, c and d .

v	(a, b, c, d)	$(f(a + 1, b + 1), f(c + 1, d + 1))$	$\iota(v)$
$(0, 0)$	$(0, 0, 0, 0)$	$(4, 4)$	40
$(1/\sqrt{2}, 0)$	$(0, 1, 0, 0)$	$(14, 4)$	77
$(0, 1)$	$(0, 0, 1, 0)$	$(4, 7)$	73

Table 2.2. Example of indices $\iota(v)$ for nodes v given in coordinates $(a + b/\sqrt{2}, c + d/\sqrt{2})$.

In Table 2.2 some examples of representations of nodes are given. We first add 1 to a, b, c and d , to be sure that we apply f to non negative integers. Then we apply f two times: the first time on $(a + 1, b + 1)$ and on $(c + 1, d + 1)$, and the second time on the results of the first calculations. The node $v = (a + b/\sqrt{2}, c + d/\sqrt{2})$ is then represented by an integer given by

$$\iota(v) := f(f(a + 1, b + 1), f(c + 1, d + 1)). \quad (2.5)$$

As f is invertible, for each non-negative integer $i \in \mathbb{Z}_{\geq 0}$, we can calculate the coordinates $\iota^{-1}(i)$ by applying g two times on i (see Table 2.3).

i	$g(i)$	$(g(g_x(i)), g(g_y(i)))$	$\iota^{-1}(i)$
0	$(0, 0)$	$(0, 0, 0, 0)$	$(-1 - 1/\sqrt{2}, -1 - 1/\sqrt{2})$
10	$(4, 0)$	$(1, 1, 0, 0)$	$(0, -1 - 1/\sqrt{2})$
20	$(0, 5)$	$(0, 0, 0, 2)$	$(-1 - 1/\sqrt{2}, \sqrt{2})$

Table 2.3. Integers i represent nodes in the grid \mathbb{L} .

Chapter 3

Algorithm for Reconstructing Muqarnas

We focus on the analysis of the muqarnas structure directly from its design. Our task is to find the set of elements the muqarnas consists of, together with the position of these elements. The different steps we perform to achieve this are described in this chapter. The conversion of these steps into a computer program and the visualization of the set of elements is treated in the next chapter (Chapter 4).

In Section 3.1 we define a directed graph derived from the muqarnas plan, which contains the main structure information of the muqarnas. From this graph we can recognize the muqarnas elements and their positions. The process of calculating the muqarnas is then split in two steps: first we find the directed graph containing the structure information of the muqarnas from the muqarnas plan (see Sections 3.2 and 3.4). Secondly, we extend this graph into a three-dimensional muqarnas structure (see Section 3.3).

We discover that different steps of the algorithm contain some freedom. This result in different muqarnas reconstructions with the same plane projections. The differences between such muqarnas reconstructions are discussed in Section 3.5.

3.1 Representation of the Muqarnas Structure in a Graph

For converting a muqarnas plan into a three-dimensional muqarnas vault, we want to identify for each polygon in the muqarnas plan the corresponding element, together with the position and orientation of the element. We recall from Section 2.2 that an element can be specified by its diagonal, angle, and its type (cell or intermediate element). To determine its type, we decode which edges of the polygon correspond to the curved sides of the element. If we are acquainted with the curved sides, we can specify the central node (see Definition 2.2.1). The direction of the curves according to the central node fixes then the type of the element. We will see that also the height information can be assigned from knowing all directions of the curved sides in the plan. As the plane position is already included in the coordinates of the central node, the position and orientation of the element can then be determined.

3.1.1 Definition of the Muqarnas Graph

In the left part of Figure 3.1 a cell and an intermediate element are drawn with red arrows pointing to the apex of the curved side. In the plan $P(N, E)$, the edges which are projections of curved sides appear as red arrows and define a directed subgraph $G(M, C)$ of $P(N, E)$. The blue arrows in the cell indicate the required direction of curved sides of elements in the previous tier on which the cell can be arranged.



Figure 3.1. On the left: cell and intermediate element with arrows on the curve in upward directions. On the right: corresponding plane projections.

Definition 3.1.1 (muqarnas graph). A muqarnas graph is a directed subgraph $G(M, C) \subset P(N, E)$ such that an edge $e \subset E$ appears in C iff it is the projection of a curved muqarnas side or the projection of a backside of a full cell at the bottom of the muqarnas.

If $\text{edge}(v, w) \in C$ correspond to the projection of a curved side with w the projection of the apex of the curve then the $\text{edge}(v, w)$ is directed so that it points from v to w . If $\text{edge}(v, w)$ is the projection of a backside of a cell then $\text{edge}(v, w)$ points to w with w the projection of the vertex joining to the curved side of the cell. We write $\text{arw}(v, w)$ to refer to an arrow in the muqarnas graph pointing from v to w .

The muqarnas graph mainly consists of the projection of the curved element sides. Beside that, the projection of the backsides of bottom cells are included in the muqarnas graph. In this way the structure information in which way the muqarnas can be continued downwards is obtained. If we add elements at the bottom of the muqarnas, the backsides of the bottom cells can only stand on the curved sides of the new elements in the direction as fixed by the muqarnas graph.

The arrows in the interior of the graph G can correspond to one or two curved sides. At this place elements on the same tier or of different tiers can join. At the left of Figure 3.2 the red arrow is the projection of two curved sides (curve to curve), in the right figure the red arrows are projections of a backside of an element and a curve side of an element in the lower tier (back on curve). The projection of the place where a backside of a cell stands on the front of an intermediate element (back on front) and the projection of the place where the front of an intermediate element touches the front of an intermediate element (front to front) do not appear in the muqarnas graph.

In Figure 3.3 we see an example of the relationship between a muqarnas plan P and the corresponding muqarnas graph G . In this example the graph G is a proper subgraph of the plan P , as there are lines in the plan that do not correspond to curved sides and

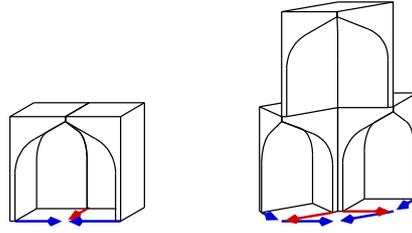


Figure 3.2. Projections of curved sides may correspond to two joining curved sides of the same tier (left), or to one curved side of an element and a backside of an element in the overhead tier.

therefore not appear in the graph G . Both objects, a muqarnas plan and a muqarnas graph, are graphs if we consider them as mathematical objects. A muqarnas plan is an undirected graph and a muqarnas graph is a directed graph. To avoid confusion, we will use the term graph only for the directed graphs and use the term plan to refer to an undirected graph.

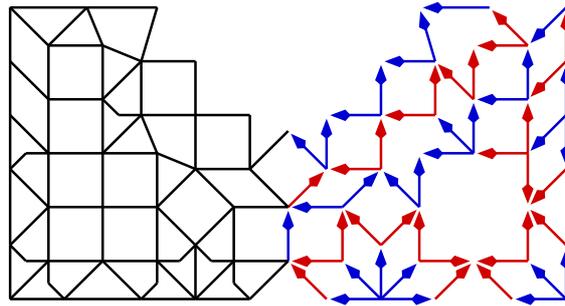


Figure 3.3. Muqarnas plan and graph of the muqarnas vault in the basement vault of the north iwan in the Friday Mosque in Natanz (see Figure 1.1(c)).

3.1.2 Properties of the Muqarnas Graph

The muqarnas graph is defined in terms of the three-dimensional muqarnas structure. Our aim is to reconstruct the muqarnas directly from its design. For this reason, we need to recognize the muqarnas graph in the plan without using other information of the muqarnas vault as that contained in the plan. In this section we formulate some properties of the muqarnas graph so that we can clear what kind of object we want to create.

From the property of a muqarnas structure that the elements are arranged in tiers, it follows that the length of the paths in a muqarnas graph with same end-nodes is the same. To formalize this statement, we first introduce the terminology of height for nodes in a muqarnas graph $G(M, C)$.

Definition 3.1.2. Let $G(M, C)$ be a muqarnas graph. We define the height $h(v) \in \mathbb{Z}_{\geq 0}$ of a node $v \in M$ as follows: if v appears as the projection of a vertex at the backside of a muqarnas element in tier r , which belongs to the bottom of the muqarnas, then $h(v) = r - 1$. Otherwise $h(v)$ is given by the number of the minimal tier containing elements with node v in their projection.

We recall from Section 2.1.2 that following the direction of a curve in the muqarnas, corresponds to going upwards. This means that every arrow $c \in C$ of the graph $G(M, C)$ corresponding to a curved side in tier r points to tier $r + 1$. This property will later be used for calculating the heights of the nodes directly from the muqarnas graph.

Lemma 3.1.3. Let $G(M, C)$ be a muqarnas graph. For the height $h(v)$ of the nodes $v \in M$ the following properties are valid:

$$h(w) = h(v) + 1 \text{ if } \text{arw}(v, w) \in C \quad (3.1)$$

and

$$\min_{v \in M} h(v) = 0. \quad (3.2)$$

PROOF. Let the arrow $\text{arw}(v, w)$ be the projection of the curved side of an element in tier r . Then w is the projection of the apex of this curve. It cannot touch curved sides of elements in a lower tier and therefore $h(w) = r$. The node v corresponds to a vertex on the bottom of the curve. This vertex belongs also to the backside of the element. If the element belongs to the bottom of the muqarnas, it follows directly from Definition 3.1.2 that $h(v) = r - 1$. Otherwise, the element stands on an element of the previous tier. If the element is a cell, its backsides stand on curved sides of an element below (back on curve), or its backsides stand on the front of an element below (back on front). In both situations the bottom of the element touches elements a tier lower. We find that v also appears as the projection of an element in the lower tier and $h(v) = r - 1$.

If the curved side of which $\text{arw}(v, w)$ is the projection belongs to an intermediate element not at the bottom of the muqarnas, then this element stands with its backside on the front of an element in the previous tier. In that case we also find that $h(v) = r - 1$.

For (3.2), we remark that this follows directly from Definition 3.1.2. The height $h(v)$ cannot be negative and the projection of vertices of the bottom of the elements in the first tier appear as nodes v in the muqarnas graph with $h(v) = 0$.

□

Lemma 3.1.4. If G is a directed graph corresponding to a muqarnas structure, then paths p, q in G with the same end-nodes have the same length. In other words, if p, q are paths in G with lengths $l(p), l(q)$ respectively and end-nodes: $p_s = q_s$ and $p_e = q_e$, then

$$l(p) = l(q).$$

PROOF. We remark that following an arrow corresponds in the muqarnas to going upwards to the next tier. This means that if we start at node v and we are able to walk to w ,

we need to pass $h(w) - h(v)$ tiers, with $h(v)$ the height of node v . Therefore, the length of our walk equals $h(w) - h(v)$. It depends only of the end-nodes of the path and not of the route. □

From the previous property it follows that a muqarnas graph does not contain cycles. A path starting in a node v and ending in the same node v has always length 0.

After formulating properties of the height $h(v)$, we now concern on the calculation of $h(v)$. The process of calculating the height is based on determining the distance of the nodes to the boundary. Let $|M|$ be the number of nodes in the graph $G(M, C)$. The notation M_h is used for the set of nodes $v \in M$ with $h(v) = h$. For an arrow $c \in C$ we denote c_s for its initial point and c_t for its end-point. We use the following algorithms to calculate the height of the nodes $v \in M$:

A-3.1

Initial situation: For all $v \in M$ we set $\tilde{h}(v) = 0$, $h = 0$ and $\text{change} = \text{false}$.

Iteration step: For all $c \in C$ with $c_s \in M_h$: if $\tilde{h}(c_e) < h + 1$ we set $\tilde{h}(c_e) = h + 1$ and mark the situation as changed by setting the boolean $\text{change} = \text{true}$. If this is done for all $c \in C$, we set $h = h + 1$.

Stop condition: If $h > h_{max}$ or $h > |M|$ with $h_{max} = \max_{v \in M} \tilde{h}(v)$, we return the value of change .

A-3.2

Initial situation: We set $h = h_{max}$, $\text{change} = \text{false}$.

Iteration step: For all $c \in C$ with $c_e \in M_h$: if $\tilde{h}(c_s) < h - 1$ we set $\tilde{h}(c_s) = h - 1$ and mark the situation as being changed by setting $\text{change} = \text{true}$. If done for all $c \in C$ we set $h = h - 1$.

Stop condition: We stop if $h = 0$ and return the value of change .

A-3.3

Initial situation: We set $k = 0$, $\text{change} = \text{false}$.

Iteration step: Apply A-3.1 and A-3.2 and set $k = k + 1$. If A-3.1 or A-3.2 returns true we set $\text{change} = \text{true}$.

Stop: If $\text{change} = \text{false}$ or if $k = |M|$.

The main algorithm is A-3.3. This algorithm calls the algorithms A-3.1 and A-3.2 in alternating order. The Algorithm A-3.1 forces that all successors of the nodes at height h get height $h + 1$ and the Algorithm A-3.2 forces the predecessors at height h having height $h - 1$. In Figure 3.4 we see a part of the muqarnas graph from the interpretation of Harb of the design on the plate found at the Takht-i-Sulaymān. On the left we see the result of calculating the heights of the nodes by only applying Algorithm A-3.1 and on the right in red the changes after applying Algorithm A-3.2. By applying the Algorithms A-3.1 and

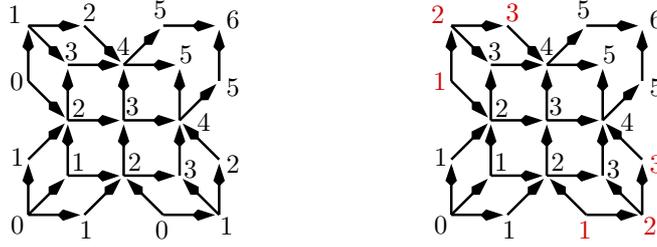


Figure 3.4. Part of a muqarnas graph. On the left: calculated heights of the nodes after running A-3.1. On the right: calculated heights of the nodes after running additionally A-3.2.

A-3.2 in alternating order, we correct the height information needed for the nodes not direct reachable by a path in the graph starting at height 0. We force the algorithm to stop if $h_{max} > |M|$. This is needed for the situation we apply the algorithm on a graph which is not a muqarnas graph. In that case it may contain a cycle and applying the algorithm could result in an infinite loop. Applying the algorithm on graphs different from muqarnas graph can be useful to test whether a given graph may correspond to a muqarnas or not. The following lemma affirms that the height is calculated well by Algorithm A-3.1.

Lemma 3.1.5. *If $G(M, C)$ is a connected muqarnas graph then after running the Algorithm A-3.3, for each node $v \in M$, the calculated height $\tilde{h}(v)$ equals the height $h(v)$ of the node v as defined in Definition 3.1.2.*

PROOF. We write $\tilde{h}_k(v)$ for the calculated height $\tilde{h}(v)$ after the iteration step k in Algorithm A-3.3. The Algorithms A-3.1 and A-3.2 cause only a change of the actual value $\tilde{h}(v)$ for a node v if this results in an increase. We conclude that

$$\tilde{h}_{k+1}(v) \geq \tilde{h}_k(v). \quad (3.3)$$

We define the sequence v_0, \dots, v_k given by the history of $\tilde{h}_k(v_k)$. With this we mean that v_i is the node which determines the value of $\tilde{h}_{i+1}(v_{i+1})$ in the iteration step $i + 1$. The following situations can appear:

1. $\tilde{h}_{i+1}(v_{i+1}) = \tilde{h}_i(v_i)$: then $v_{i+1} = v_i$, there was no change in the iteration step $i + 1$.
2. $\tilde{h}_{i+1}(v_{i+1}) = \tilde{h}_i(v_i) + 1$: there is an arrow $\text{arw}(v_i, v_{i+1})$, the Algorithm A-3.1 causes a change in the iteration step $i + 1$.
3. $\tilde{h}_{i+1}(v_{i+1}) = \tilde{h}_i(v_i) - 1$: there is an arrow $\text{arw}(v_{i+1}, v_i)$, the Algorithm A-3.2 causes a change in the iteration step $i + 1$.

These situations compared with Lemma 3.1.3 show that in all three cases

$$\tilde{h}_{i+1}(v_{i+1}) - \tilde{h}_i(v_i) = h(v_{i+1}) - h(v_i).$$

From this it follows that

$$\tilde{h}_k(v_k) - \tilde{h}_0(v_0) = \sum_{i=0}^{k-1} [\tilde{h}_{i+1}(v_{i+1}) - \tilde{h}_i(v_i)] = \sum_{i=0}^{k-1} [h(v_{i+1}) - h(v_i)] = h(v_k) - h(v_0).$$

As $\tilde{h}_0(v_0) = 0$, we conclude that

$$\tilde{h}_k(v_k) \leq h(v). \quad (3.4)$$

We proceed the proof by showing that for all nodes v there is a $k < |M|$ such that $\tilde{h}_k(v) = h(v)$. Because $\tilde{h}_k(v)$ is increasing in k (3.3) and smaller than or equal to $h(v)$ (3.4), we can then conclude that

$$\tilde{h}(v) = h(v).$$

There is at least one node $v^* \in M$ with $h(v^*) = 0$. For this node already $\tilde{h}_0(v^*) = 0$ and thus $\tilde{h}(v^*) = h(v^*)$. If the height of a node v is set well: $\tilde{h}_k(v) = h(v)$ then in the next iteration step the height of its neighbors are set well: Algorithm A-3.1 will care about that $\tilde{h}_{k+1}(w) = h(w)$ for all successors w of v and Algorithm A-3.2 will force $\tilde{h}_{k+1}(w) = h(w)$ for all predecessors w of v .

We define the plan $P(M, E)$ given by the graph $G(M, C)$ as the undirected graph for which $\text{edge}(v, w) \in E$ if $\text{arw}(v, w) \in C$ or $\text{arw}(w, v) \in C$. Let v be such that $\tilde{h}_k(v) = h(v)$. Because the graph G is connected, we find that for all nodes $w \in M$, there exist a path in the plan P from v to w . Let this path be given by the sequence $v = v_0, v_1, \dots, v_n = w$. Then $\text{arw}(v_i, v_{i+1})$ or $\text{arw}(v_{i+1}, v_i)$ exist in G . With induction we can conclude that after at most $k + n$ iteration steps also $\tilde{h}_{k+n}(v) = h(v)$. As the shortest path has length smaller than $|M|$ the height of all nodes are given in less than $|M|$ steps. □

The next feature, which we will formulate in Lemma 3.1.7, concerns about the direction of the edges in the muqarnas graph. To formulate this lemma, we need the notion of *opposite edges* in a graph. In the definition of opposite edges not only the muqarnas graph, but also the plan is involved. The definition is illustrated in Figure 3.5.

Definition 3.1.6 (opposite edges). *We consider a muqarnas with plan $P(N, E)$ and muqarnas graph $G(M, C)$. If $e = \text{edge}(e_s, e_t)$ and $f = \text{edge}(f_s, f_t)$ are edges in the graph $G(M, C)$ such that there exist edges $e' = \text{edge}(e_s, f_s)$, $f' = \text{edge}(e_t, f_t)$ in $G(M, C)$ but no cross edges $\text{edge}(e_s, f_t)$ or $\text{edge}(f_s, e_t)$ in $P(N, E)$, then we say that e and f are opposite edges in G according to P .*

In other words: edges e and f are opposite if both end-nodes of e are connected in the graph G as well as in the plan P with exactly one, and not the same, end-node of f . In this definition of opposite, the end-nodes of the edges need to be connected in the graph G for being opposite. To exclude the possibility to be opposite if these end-nodes have a second connection we look at the plan P . In Figure 3.5 the edges which appear in P , but not in G are dotted. The edges drawn by continuous lines appear both in the plan P as well as in the subgraph G . We see three different kind of structures which may appear

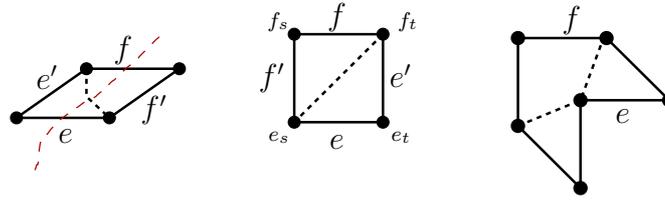


Figure 3.5. On the left: edges e and f are opposite. In the middle and on the right: edges e and f are not opposite.

in the graph. In the first picture, edges e and f are opposite, because the end-nodes of e are connected in the graph G to the end-nodes of f . In the second picture e and f are not opposite, as the end-node e_s is connected in the plan P to both end-nodes of f . In the last picture e and f are not opposite, as not both end-nodes of e are connected in the graph G to the end-nodes of f .

Lemma 3.1.7. *We observe a muqarnas with graph G and plan P . If the edges e and f are opposite in the graph G according to the plan P , then they have the same direction in the graph G .*

PROOF. We look at the polygon at the left of Figure 3.6. We consider this polygon as a part of the graph G with opposite edges e and f . As this polygon appears in the graph G , we have to determine for all edges a direction. The upper four pictures represent the situation that this polygon corresponds to a cell. In that case the edges f and f' join in their end-points, represented by black arrows in the upper four pictures. The red arrows present all possible directions of the remaining edges e and e' . Situation I_a may appear in muqarnas graphs, the backsides of the corresponding cell could stand on curved sides of elements in the lower tier. In this situation the opposite edges e and f have indeed the same direction. Picture II_a presents a situation which is in contradiction with Lemma 3.1.4: there is a path of length 1 and a path of length 3 both with the same end-nodes. Picture III_a implies that f and f' as well as e and e' define a cell. We need to join these cells with their backsides. This situation does not appear in muqarnas and therefore we do not find situation III_a in muqarnas graphs.

The lower four pictures show the situation the left polygon corresponds to an intermediate element. In that case the edges e and e' join in their initial points and present the curved sides of this intermediate element (black arrows). The first situation I_b may appear in a muqarnas graph G . In this situation a cell stands with its backsides on the front of the intermediate element. The projection of the front of the intermediate does not appear in the graph. The edges e and f have in this case the same direction. The situation II_b is in contradiction with Lemma 3.1.4. For the third situation, III_b , we remark that this situation only appears when e and e' define a half intermediate element and f and f' also define a half intermediate element. These half intermediate elements appear on the same tier and touch in their front parts. This can only happen in case of half intermediate elements. In that case the projection of the front is a cross edge joining

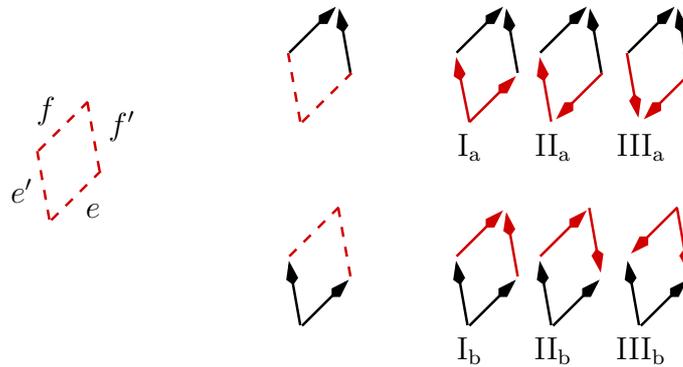


Figure 3.6. Possible directions of the edges of the polygon given by e, e', f and f' . If edges e and f are opposite edges in a muqarnas graph then only cases I_a and I_b are allowed.

the end-points of the edges e and e' in the plan P . This is in contradiction with the assumption of e and f being opposite edges.

We conclude that the only allowed situations are I_a and I_b , in those cases e and f have the same direction.

□

3.2 Construction of Muqarnas Graphs from the Plan

In Definition 3.1.1 we defined the muqarnas graph as the projection of the curved element sides of the muqarnas. To construct the muqarnas from its design, we should find from the plan the directed subgraphs with the same properties as muqarnas graphs. We facilitate the problem by considering first the situation where the plan only consists of edges appearing in the muqarnas graph. The starting-point is already a subplan and our task is to determine the directions of its edges.

We have seen in Lemma 3.1.7 that opposite edges of the muqarnas graph G , according to the plan P , have the same direction. Therefore, we bind opposite edges together in one *orbit* (see Figure 3.7) and we determine for each orbit a direction instead of determining separately for each edge a direction. An orbit is defined as an equivalence class of the equivalence relation generated by the opposite edges. Edges of the same equivalence class are then called *parallel edges*.

Definition 3.2.1 (parallel edges). *The edges e and f are parallel if $e = f$, or if e and f are opposite, or if there exist an edge parallel to both e and f .*

We can prove that this definition indeed defines an equivalence class by verifying the reflexive, symmetric and transitive relations.

Definition 3.2.2 (orbit). *An orbit is an equivalence class of the equivalence relation given by the parallel edges.*

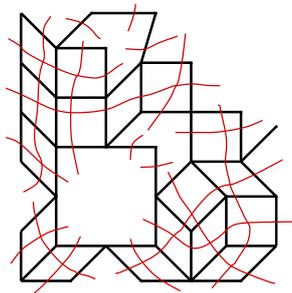


Figure 3.7. The colored lines represent the orbits in the muqarnas graph.

An arrow in the muqarnas graph corresponds to a curved side. Following the direction of an arrow in the graph means in the muqarnas going to the next tier. Our job is to set the direction of the edges in such a way that we can walk over the arrows from the bottom boundary to the center. In the practise, a muqarnas is built from the outside in (see [Ibrahim, 2002, pp. 77–78]). Hence, we read a muqarnas design in this direction. We formulate the following rules:

1. Arrows at the bottom boundary need to point inward.
2. Arrows at the center, need to point to the center.

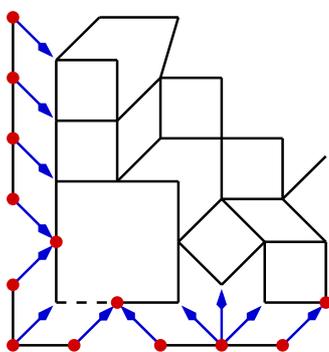


Figure 3.8. The red nodes are the bottom boundary nodes of the plan. Rule 1 should be able to set the directions of the blue arrows. The direction of the dotted edge is undetermined.

In Figure 3.8 we see the desired result after applying the first rule. Not for all edges incident to a bottom boundary node the direction can be determined. For example, the direction of the dotted edge cannot be set. We can only set the direction of edges of which we can validate which end–node is the projection of the lowest vertex. Not all bottom boundary nodes necessarily appear as the projection of a more lower vertex than its neighbors. For the dotted edge in Figure 3.8, the end–node belonging to the bottom

boundary is the projection of vertices above the vertices with projection the end–node not in the bottom–boundary.

We distinguish two kinds of bottom boundary nodes, the *singular* and the *non–singular* ones. They are so defined that the height of the non–singular nodes is always smaller than the height of its neighboring nodes not on the bottom boundary. For the singular nodes we cannot do any statement about this.

We define the line $l(u, w)$ from u to w by

$$l(u, w) := \{tu + (1 - t)w : t \in \mathbb{R}\},$$

and the line section from $l[u, w]$ from u to w by

$$l[u, w] := \{tu + (1 - t)w : t \in [0, 1]\}.$$

We say that the line $l(u, w)$ *separates* v from the plan $P(N, E)$ if and only if there is a line section $l[v, v']$ with $v' \in N_{-}\{v \cup l(u, w)\}$ so that $l(u, w) \cap l[v, v'] \neq \emptyset$ (see Figure 3.9).

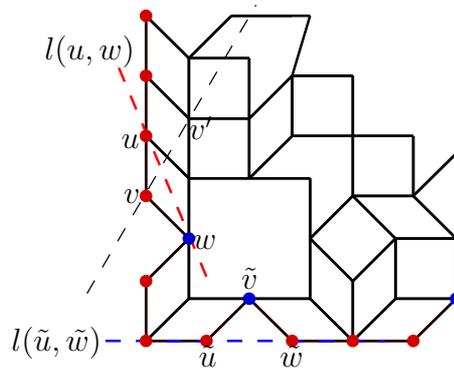


Figure 3.9. Node v is non–singular, as it is separated from P by $l(u, w)$. Node \tilde{v} is singular, the line $l(\tilde{u}, \tilde{w})$ does not separate it from the plan P .

Definition 3.2.3 ((non–)singular). Let $v \in W_{bb}(P)$ with neighbors u, w in $W_{bb}(P)$. If $l(u, w)$ does not separate v from P and $v \notin l(u, w)$ then we call v a *singular bottom boundary node*. If $v \in l(u, w)$ or $l(u, w)$ separates v from P then v is *non–singular*.

The set of non–singular boundary nodes of the plan P is denoted by $\overset{\circ}{W}_{bb}(P)$. The first rule can now be applied to set the directions of edges pointing from the non–singular bottom boundary nodes to other nodes of the plan:

Rule 1. We consider a muqarnas structure with plan $P(N, E)$ and graph $G(M, C)$. Let $v \in \overset{\circ}{W}_{bb}(P)$ and w adjacent to v with $w \notin \overset{\circ}{W}_{bb}(P)$, then $\text{arw}(v, w) \in C$.

The second rule is motivated by the fact that paths over curved sides finish at the front or in the center of the muqarnas. From the center, it is not possible to go upwards without leaving the center.

Rule 2. We consider a muqarnas structure with plan $P(N, E)$ and graph $G(M, C)$. If the corresponding muqarnas has a regular center and $c \in C$ has end-nodes c_s, c_t such that $c_s \notin W_c(P)$ and $c_t \in W_c(P)$, then $c = \text{arw}(c_s, c_t)$.

Rule 2 can only be applied if the center is regular, that means all center nodes appear as projections of vertices of the upper most tier. This is usually the case, the only example we know where this is not true are several interpretations of the design on the plate found at the Takht-i-Sulaymān (see Section 5.2.5). In Figure 3.10 a part of the center of a possible muqarnas graph corresponding to the design on the plate is drawn. If we

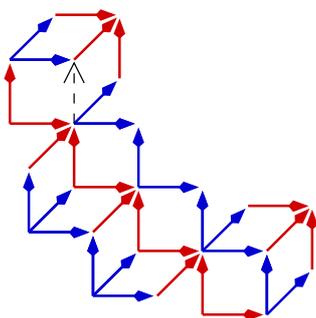


Figure 3.10. Part of the center of a muqarnas graph, inspired by the design found at Takht-i-Sulaymān. The dotted arrow has a direction which is in contradiction with Rule 2.

observe the dotted arrow, we see that for its end-nodes $c_s \in W_c(P)$ and $c_t \notin W_c(P)$, but $\text{arw}(c_t, c_s) \notin C$, which is in contradiction with Rule 2. In this special case node c_s belongs to the center, but not to an element in the upper most tier. This makes it possible to walk from c_s further upwards to node c_t , leaving the center. Because we doubt whether the center is designed well (see [Dold-Samplonius and Harmsen, 2005]), and this is the only example we found of muqarnas without regular center, we do not consider this situation as being representative. In the case we allow a non-regular center, we replace this rule by an alternative one which is a generalization of Rule 2.

Rule 2 (alternative). We consider a muqarnas structure with plan $P(N, E)$ and graph $G(M, C)$. If $c \in C$ belongs to an orbit, which does not contain an edge e with end-nodes $e_s, e_t \in W_c(P)$ and for the end-nodes c_s, c_t of c it applies that $c_s \notin W_c(P)$, $c_t \in W_c(P)$ then $\text{arw}(c_s, c_t) \in G(M, C)$.

A third rule for setting the direction of the edges, can be formulated by recognizing some known structures. These are the figures in the plan appearing after removing edges in the plan. By removing edges there arise islands, as defined in Definition 2.2.7. In Figure 3.11 an overview of such kind of figures is given, together with the direction of the edges. They only appear in the orientation as given in Figure 3.11.

These figures correspond to more than one element in the muqarnas, and will consist of one cell and different intermediate elements. The projection of the curved sides of the

cell is in Figure 3.11 given by red arrows and the projection of the curved sides of the intermediate elements with blue arrows. Remark that the arrows of all these figures have

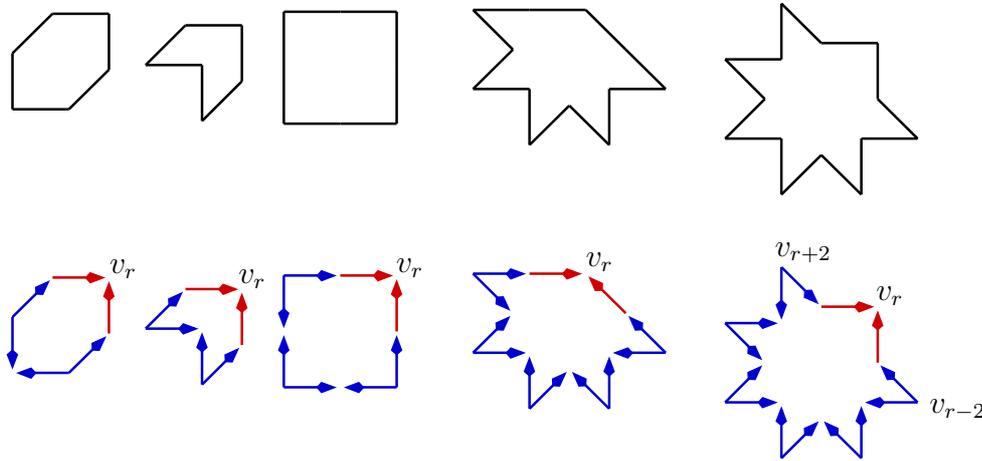


Figure 3.11. Overview of different islands with their common directions in the graph.

the same structure: at the most right node on the diagonal, two arrows come together. The other arrows have an alternating direction so that the last arrow of the alternating ones points to the arrow pointing to the most right diagonal node.

In formulating this rule we use the term *sink* for a node in a directed graph which is an initial point for all arrows incident to this node. Similar a *source* is a node which is an end-point for all arrows incident to it. We remark that an island always consists of an odd number of nodes, as each element is represented by two arrows in the graph.

Rule 3. Let $\{v_0, \dots, v_n\}, v_0 = v_n$ be a counter-clockwise oriented cycle in the subplan Q representing an island in a muqarnas graph G . Let v_r be the diagonal node of this cycle in G with largest x -coordinate, then $\text{arw}(v_{r-1}, v_r), \text{arw}(v_{r+1}, v_r) \in G$ and for $i \neq 2, \dots, r$ and $j = 2, \dots, n - r$ the nodes v_{r-i}, v_{r+j} are sources in this cycle for i, j even and sinks in the cycle for i, j odd.

We formulate a fourth rule which excludes some forbidden element combinations. In the left picture of Figure 3.12 we see a situation where node u is the central node for all intermediate elements containing u in its plane projection. In this example the only possibility to add new elements in the next tier, is to put cells with their backsides on the fronts of the intermediate elements (see Section 2.1.2). The next tier will then consist of cells spread over the whole circumference of the structure as given in this picture. In this way a column appears, a structure not fitting in our niche or dome. We see that node u cannot be a source in the muqarnas graph.

In the second picture node v is the central node of all cells which have v in its plane projection. In the muqarnas this implies that all curved sides join already to other curved sides in this structure and no free curved sides are left. In Section 2.1.2 we have seen

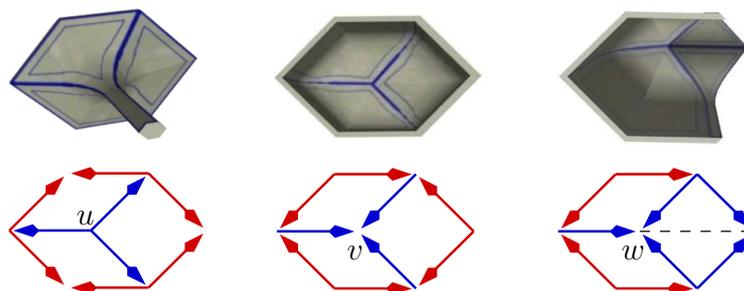


Figure 3.12. First two pictures show element combinations not allowed in a muqarnas fitting in a dome or niche. The right picture shows a similar structure, with a combination of elements which is allowed.

that the only way cells can join to other elements is by: joining curved sides at the same tier together (curve to curve), backsides of the cells stand on front sides or curved sides of elements in the previous tier (back on front or back on curve). This means for the situation as given in the second picture that it is not possible to join elements in the same tier, or to join a new tier above. Such a structure therefore can only appear in the center of a muqarnas. We exclude this situation by Rule 4, by avoiding sinks in the graph.

In the right picture node w is also a sink, but here it is possible to add new elements. In node w not only cells join, but also half intermediate elements. There are still free curved sides available to which we can join new elements. We summarize our observations in Rule 4:

Rule 4. *Let v be a node in a muqarnas plan $P(N, E)$, then v cannot be a source in the muqarnas graph. If all edges $e \in E$ incident to v appears as arrows in the corresponding muqarnas graph G , then node v cannot be a sink in the corresponding muqarnas graph.*

We have now a set of rules available which make it possible to determine the directions of most edges in the muqarnas graph, by using only the plane projection of the muqarnas, and not the muqarnas vault itself. The definition of opposite edges and the rules given in this section are formulated in accordance with the different element combinations appearing in Il-Khanid and Seljuk muqarnas, fitting in a niche or dome. If we restrict ourselves to the reconstruction of Il-Khanid muqarnas, we can simplify our definition of opposite. In Il-Khanid muqarnas two half intermediate elements fitting at their front do not appear. We therefore do not need to exclude the situation as presented in the middle picture of Figure 3.5. Developing reconstruction algorithms for other types of muqarnas may need a reformulation of the rules formulated in this section.

In Chapter 5 we will see that with this set of rules most directions of orbits can be set. There remain only a few undetermined orbits and we can calculate for all possible graphs the corresponding muqarnas reconstruction. Often the different directions of these undetermined orbits result in different valid muqarnas computer reconstructions.

3.3 Conversion of Muqarnas Graph into Muqarnas Structure

In the previous chapter, we interpreted a muqarnas structure as a set of elements. These elements are given by their type and their position. In this section we explain how the muqarnas structure can be found from the muqarnas graph. In Section 3.3.1 we give an overview of the whole conversion process. This process is split in two parts. First, we determine which nodes together define a muqarnas element (see Section 3.3.2). After that, these combinations of nodes are converted into a set of muqarnas elements combined with their position (see Section 3.3.3). In Section 3.3.4 we show that it is in some situations possible to find the muqarnas structure directly from the muqarnas graph, without using the muqarnas plan.

3.3.1 Muqarnas Reconstruction Process

We recall from Section 2.2.1 that if two arrows in the muqarnas graph join in their end-points, these two arrows define a cell. Intermediate elements are given in the graph by two arrows joining in their initial points. Therefore, we are interested in joining arrows to find the elements of the muqarnas. The way these arrows join decodes the type of the element. Intermediate elements are required to fill in the gaps in the view from underneath. Therefore, two arrows starting from the same node only represent an intermediate element if there will appear a gap otherwise.

We use the word *face* to refer to a set of nodes which corresponds to an element. More formal we define a face as follows:

Definition 3.3.1 (face). A face $F(c, v, w)$ in a muqarnas graph $G(M, C)$ is a set of three nodes $c, v, w \in M$ such that the edges $\text{edge}(v, c)$, $\text{edge}(w, c)$ correspond to the curved sides of the same element in the muqarnas structure.

A face defines a figure (see Definition 2.2.6) in the plan P by ordering the nodes c, v, w adequate. The node c is the central node of the corresponding element, since this is the place where the edges $\text{edge}(v, c)$ and $\text{edge}(w, c)$ join. In the case that the face $F(c, v, w)$ corresponds to a cell we call it a *cell-face*. A cell-face is represented in the muqarnas graph by arrows $\text{arw}(v, c)$ and $\text{arw}(w, c)$ (see Figure 3.13 on the left). Similar, if it corresponds to an intermediate element we call it an *int-face*. In the muqarnas graph the int-face $F(c, v, w)$ is represented by the arrows $\text{arw}(c, v)$ and $\text{arw}(c, w)$ (see Figure 3.13 on the right).

The algorithm for finding the muqarnas structure from a muqarnas graph is split in five steps, where the first three steps concerns about setting the faces and the last two convert the faces into the elements defining the muqarnas structure:

1. Find the cell-faces corresponding to full cells.
2. Find the int-faces.



Figure 3.13. Set $\{c, v, w\}$ defines a face. On the left: cell-face. On the right: int-face.

3. Find the cell-faces corresponding to half cells.
4. Translate cell-faces to cells.
5. Translate int-faces to intermediate elements.

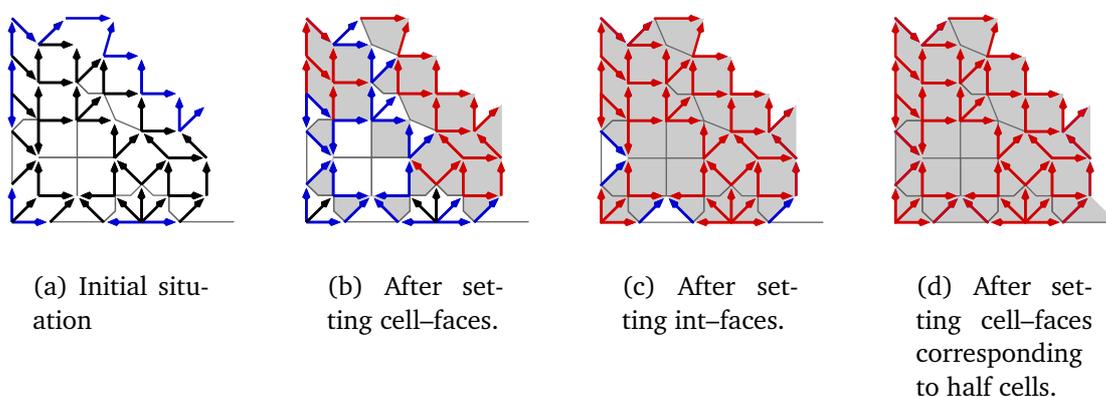


Figure 3.14. Finding the faces from the muqarnas graph corresponding to the muqarnas in the basement of the north iwan of the Friday Mosque in Natanz. Faces already set are colored gray. Black arrows are twice available for setting faces, blue arrows are once available, and red arrows are not available any more.

In Figure 3.14 the initial situation and the first three steps of the algorithm are visualized. We mark in each step how often the arrows are used so that after setting the cell-faces, we can decide whether there is place for an intermediate element left. To find the set of cell-faces corresponding to half elements another method is used as for finding the cell-faces corresponding to full elements. Therefore, this is done in a separate step. In the next section we explain more detailed how the faces can be recognized from the muqarnas graph.

For the administration how often the edges are used, we remark that an edge in the interior of the plan corresponds to the sides of two elements. If this edge is represented by an arrow in the graph, then it can correspond to two curved sides (curve to curve) or to one curved and to one backside of an element (back on curve). During the administration process, we therefore also need to consider the situation that edges are used as backsides.

At the boundary the edges are only one time available as curved side of an element, or as backside of a cell. We start our administration by setting all edges at the boundary as used once.

3.3.2 Determination of Faces from the Graph

In Definition 3.3.1 a face is defined as the projection of parts of elements in the muqarnas structure. Because our task is to reconstruct the muqarnas, this information from the muqarnas structure is not available. We need to recognize the faces by using the plan $P(N, E)$ and the muqarnas graph $G(M, C)$.

We find the cell-faces corresponding to full cells by looking for combinations of arrows joining in their end-nodes. For each node $c \in M$ we search predecessors v, w with a common neighbor $k \neq c$ in the plan P . This fourth node k verifies that c, v, w is part of a figure and we can set the cell-face $F(c, v, w)$. We look for the fourth vertex k in the plan P and not in the muqarnas graph G . They do not necessarily exist in the G , because $\text{edge}(k, v)$ and $\text{edge}(k, w)$ could correspond to the backside of the cell. These backsides do not appear in the graph in the situation that the corresponding cell stands with its backsides on the front sides of intermediate elements. No curved sides at this place are involved then.

To mark which arrows of the graph G are used we first select the real backsides of the cell. For setting the cell-face we were satisfied by knowing that v and w have a common neighbor k , but this common neighbor does not need to be unique (see Figure 3.15). For the backsides we choose for the common neighbor k of v and w nearest to c :

$$k^* := \operatorname{argmin} \{ \|k - c\| : k \in N, k \neq c, \text{edge}(k, v), \text{edge}(k, w) \in E \}.$$

If the arrows $\text{arw}(k^*, v), \text{arw}(k^*, w)$ exist in the muqarnas graph G , they need to be marked as used. In addition, the arrows $\text{arw}(v, c)$ and $\text{arw}(w, c)$, which correspond to the curved sides of the cell need, of course, also be marked as used.

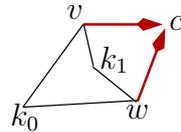


Figure 3.15. Backsides of a cell are given by the edges connecting the initial points of joining arrows.

To set the int-faces we look at the subgraph $\tilde{G}(\tilde{M}, \tilde{C})$ of the muqarnas graph $G(M, C)$ consisting of the arrows still available for defining a face. That is, we remove all arrows from C which are used twice to define \tilde{C} . The set \tilde{M} consists of the nodes incident to the remaining arrows \tilde{C} . For each node $c \in \tilde{C}$ with two or more successors v_0, \dots, v_n , we

sort its successors , so that

$$\alpha(c, v_0, v_i) < \alpha(c, v_0, v_j) \quad \text{if } i < j.$$

As we see in Figure 3.16, the int-faces are now given by $F(c, v_i, v_{i+1})$ for $i = 0, \dots, n - 1$.

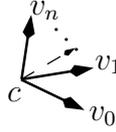


Figure 3.16. The successors v_0, \dots, v_n of c are ordered in counter-clockwise direction.

During the step of setting the int-faces, we do not search for a possible fourth node k in the plan P . In this step we find the int-faces corresponding to *full* intermediate elements as well as the int-faces corresponding to *half* intermediate elements. We need to mark the arrows $\text{arw}(c, v_0)$ and $\text{arw}(c, v_n)$, corresponding to the *outer* curved sides of the intermediate elements, as used. The arrows $\text{arw}(c, v_i)$ for $i = 1, \dots, n - 1$, which correspond to the *inner* curved sides of the intermediate elements, are used twice. The front sides of the intermediate elements do not appear in the graph G , because they cannot touch curved sides (see Section 2.1.2). Therefore, they are not considered during the administration.

We set the cell-faces corresponding to half cells at similar way as the int-faces. In this step we also iterate over the nodes $c \in \tilde{C}$ of the graph $\tilde{G}(\tilde{M}, \tilde{C})$ defined by the available edges. We scan the predecessors v_0, \dots, v_n of the node c , and find the cell-faces by sorting these predecessors by increasing angle:

$$\alpha(c, v_0, v_i) < \alpha(c, v_0, v_j) \quad \text{if } i < j,$$

like in the situation for the int-faces. We mark the arrows $\text{arw}(c, v_i)$, which corresponds to curved sides as used.

3.3.3 Conversion of Faces into Muqarnas Elements

To convert the faces into elements, we have to determine for each face its type and position. In this section $F(c, v, w)$ is a face in a muqarnas graph $G(M, C)$ so that the ordered set $\{v, c, w\}$ defines a figure in the corresponding muqarnas plan $P(N, E)$. w so

To determine the type of the element we remark that $F(c, v, w)$ is a cell-face if $\text{arw}(v, c) \in C$ and an int-face if $\text{arw}(c, v) \in C$. The diameter d of the corresponding element is given by

$$d := \min \{ \|k - c\| : k \in N, k \neq c, \text{edge}(k, v), \text{edge}(k, w) \in E \}.$$

If the set $\{ \|k - c\| : k \neq c, \text{edge}(k, v), \text{edge}(k, w) \in P \}$ is empty, there is no diagonal and the face corresponds to a half element, we set $d = 0$. The type of the element is given by

the nodes $a = v - c$, $b = w - c$ and the diameter d . The angle of the element is given by $\alpha(c, v, w)$ which equals to $\alpha(0, a, b)$. The element has curved sides of sizes $\|a\|$ and $\|b\|$.

The position of the element corresponding to the face $F(c, v, w)$ can also be calculated from the muqarnas graph $G(M, C)$. We determine the height $h(c)$ of the central node c in the graph G by applying Algorithm A-3.3. In the situation that the face represents a cell, the central node c corresponds to the apex of the curved sides. It does not touch elements of a lower tier. Therefore, the height of the cell is given by $h(c)$. The position of a cell corresponding to an cell-face $F(c, v, w)$ is then given by $\{c, h, \phi\}$ with $h = h(c)$ and $\phi = \alpha(c, c - e_0, v)$, with $e_0 = (1, 0)$.

The central node c of an intermediate element in tier r corresponds to the bottom of the curved sides of this element. This bottom stands on elements from the previous tier, or corresponds to the bottom of the muqarnas. Therefore, the height $h(c)$ equals to $r - 1$. The position of an intermediate element corresponding to the int-face $F(c, v, w)$ is given by $\{c, h, \phi\}$ with $h = h(c) + 1$ and $\phi = \alpha(c, c + e_0, v)$, with $e_0 = (1, 0)$.

3.3.4 Reconstruction from a Simplified Plan

The plate found at the Takht-i-Sulaymān shows that a muqarnas design is not always composed of the projection of all muqarnas elements (see Figure 1.3). The drawing on this plate mainly consists of squares and rhombi. We call such a plan a *simplified plan*. Some rhombi need to be split into an almond and a biped to get a proper interpretation of the design. For this reason we extend our reconstruction algorithm so that it can also handle designs only consisting of squares and rhombi. This extension is only done for muqarnas consisting of basic elements, like common in the Il-Khanid architecture. In a muqarnas where a wider variety of elements is used, we cannot determine the place the polygons can be split.

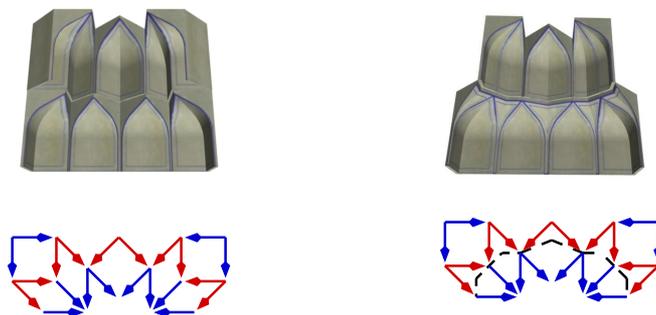


Figure 3.17. On the left: square and rhombi of the plan interpreted as one element. On the right: rhombi and center square in the plan interpreted as two elements.

In Figure 3.17 we see two different muqarnas reconstructions corresponding to a small plan based on squares and rhombi. The graphs of these interpretations are the

same, only the plans differ. These reconstructions are for illustrating, they do not correspond with real architecture. In the left picture we see a cell with a rhombus as plane projection. In this cell, the curved sides join in the smaller angle. Such elements do not appear in Il-Khanid architecture. Therefore, we interpret these rhombi as the plane projection of two elements: a cell with plane projection an almond and an intermediate element with plane projection a small biped (see Figure 3.17 on the right). The intermediate element corresponding to the biped appears one tier lower.

In similar way, the square in the middle of the plan is in the right reconstruction interpreted as a combination of a cell with plane projection a jug and an intermediate element with plane projection a large biped. The element corresponding to the biped appears one tier lower. We say that we split the cell into a combination of a cell and an intermediate element. We push down the part corresponding to the intermediate.



Figure 3.18. On the left: part of the the plane projection of a muqarnas vault in a shrine at Bisṭām (see Section 5.2.2). On the right: the plane projection of a muqarnas vault in a niche at the Friday Mosque in Natanz (see Section 5.2.1).

Splitting a cell and pushing down a part in form of an intermediate element is possible if the backsides of the cell stand on curved sides of elements one tier below (see Figure 2.13(a)). After splitting, the backsides of the cell are replaced by the curved sides of the intermediate element. Because this intermediate element is pushed down one tier, its curved sides need to join to the curved sides of the element on which the cell stood before splitting. Because splitting the elements does not influence other elements in the muqarnas structure, we call it a *local* change. Splitting the elements does not change the muqarnas graph, as the places of curved sides do not change. This means that the decision to split some elements needs in our algorithm be executed during the reconstruction of the muqarnas structure from its muqarnas graph. It does not influence the algorithm of finding the muqarnas graph from the plan.

We cannot always automatize the decision of splitting a square. If we compare in Figure 3.18 the plane projection of a vault in Bisṭām and of that in Natanz, we see that the gray marked square in the left picture is split in the muqarnas structure. In the right picture the gray marked square appears in a similar situation, but here this square is not split. These kind of decisions therefore cannot be automatized, in real life the designer need to decide about it. We restrict ourselves by automating the situation where it is

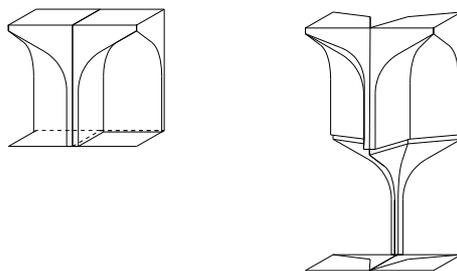


Figure 3.19. The projection of two squares needs to be interpreted as two jugs combined with a biped if the squares join at their backsides otherwise.

necessarily to split a square. This will be the situation as given in Figure 3.19: we are not allowed to join squares with their backsides together as drawn on the left. We exchange the squares by a combination of a jug and a large biped as shown on the right.

3.4 Preparation of the Plan

To determine the muqarnas graph G from the plan P , we first have to determine the subplan Q of P only consisting of the projection of the curved sides of the muqarnas elements. The muqarnas graph is then given by a directed graph based on the subplan Q . We start this section with an example to show the effect of removing edges in the plan on the corresponding reconstructed muqarnas. Then we formulate these observations so that we can use them in our software tool to find which edges we need to remove for reconstructing a muqarnas with desired properties.

We consider the muqarnas in the basement of the north iwan of the Friday Mosque in Natanz to observe the consequence of removing edges from the plan. In Figure 3.20 the complete plan corresponding to this muqarnas is drawn. Aim is to find a subplan so that there exist a muqarnas reconstruction with muqarnas graph based on this subplan and with plane projection the plan of Figure 3.20. The muqarnas consists, except at the front boundary of the vault, only of basic elements. In Section 2.1.1 we have seen that the curved sides of the basic elements always have length one. Thus, all edges corresponding to curved sides of elements, which are not located at the front of the vault, have sizes equal to the module. We therefore remove the edges of sizes not equal to the module which do not correspond to sides of elements at the front of the muqarnas. These are the edges which are marked in Figure 3.20 by dotted lines.

We apply the Rules 1–4, as formulated in Section 3.2, to calculate a possible muqarnas graph. In Figure 3.21 this graph and corresponding computer reconstruction are visualized. Comparing to the real muqarnas corresponding to this plan (see Figure 3.22), we find the following differences:

- (i). The virtual computer reconstructed muqarnas consists of 7 tiers. The real muqarnas consists of 5 tiers.

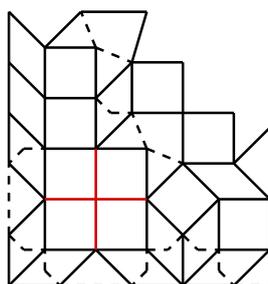


Figure 3.20. Plane projection of the muqarnas in the basement of the north iwan of the Friday Mosque in Natanz, the dotted edges have length unequal to one.

- (ii). The virtual computer reconstructed muqarnas has a non-regular bottom boundary: some nodes in the bottom boundary of the plan correspond to vertices of elements of tier 3 and 4. The real muqarnas has a regular bottom boundary, all nodes of the bottom boundary correspond to vertices on the first two tiers.

We compare the muqarnas graph of our virtual reconstructed muqarnas with the muqarnas graph of the real muqarnas. In the graph in Figure 3.21 the direction of the lower edge of the orbit marked in this graph, is given by the direction of the upper edge from this orbit as they are parallel. The direction of the upper edge is fixed by Rule 3, as this edge is a member of an island. By removing the edges marked red in Figure 3.20 we can split this orbit and the direction of the lower edges can be swapped so that we are able to reconstruct a muqarnas with a regular boundary (see the graph in Figure 3.22). We find that removing edges makes it possible to split orbits. This makes it possible to change the direction of some edges.

The edges which we need to remove are projections of non-curved sides joining to other non-curved sides. In Section 2.1.2 we have seen that non-curved sides can only join to other non-curved sides in the following situations:

1. Front sides of intermediate elements touch (see Figure 2.12).
2. Backsides of cells stand on front sides of intermediate elements (see Figures 2.13(b) and 2.13(c)).

In our example we removed the edges corresponding to situation 2 already from the plan by removing the edges with sizes unequal to the module. If the size of the backsides of the cells stand on front sides of intermediate elements equals to one, it can be more difficult to recognize such edges. We will see an example of such a situation in the next section.

We use the observations (i) and (ii) for trying to remove edges in order to calculate a subplan so that after applying Rules 1–4 there is a reconstruction possible with the desired number of tiers. Looking in our example at the edges marked red in Figure 3.20, which are the edges we need to remove, we find that removing these edges would results in an island. Islands can be created by removing all edges incident to the node in the

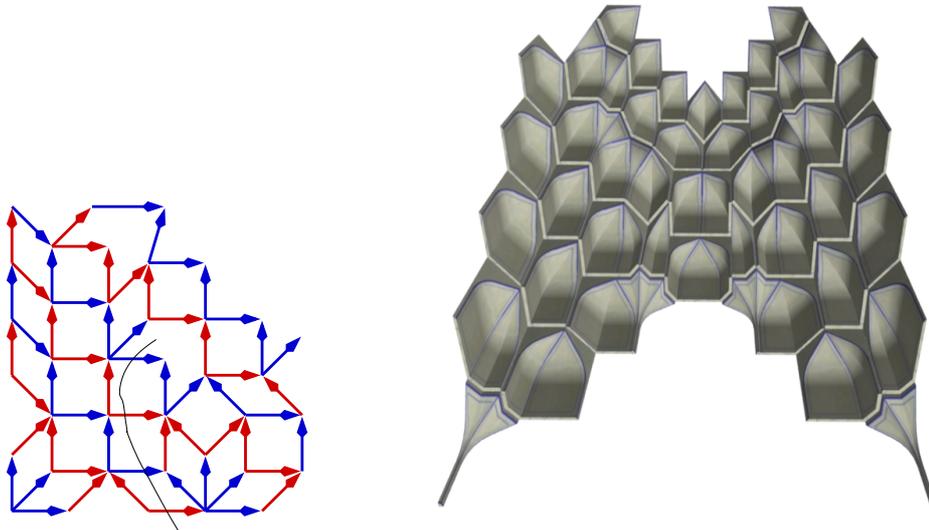


Figure 3.21. Muqarnas graph and computer reconstruction of a muqarnas with same plane projection as the muqarnas in the basement of the north iwan of the Friday Mosque in Natanz.

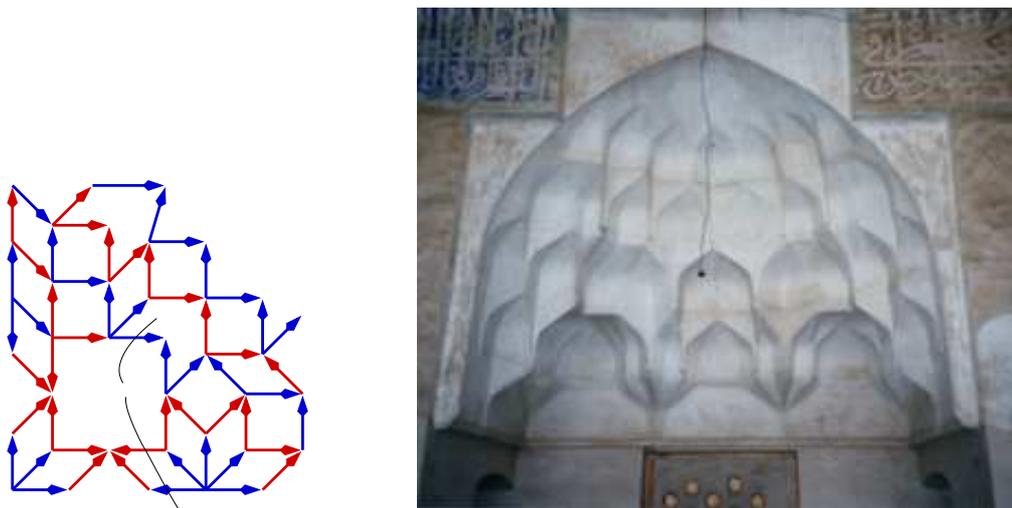


Figure 3.22. Muqarnas graph and picture of the of the muqarnas in the basement of the north iwan of the Friday Mosque in Natanz. Photo by Mohammed Bagheri.

center of the island. Islands generally appear on the diagonal of a muqarnas graph. The reason for this can be explained by looking at the paths in a muqarnas graph. The longest path usually corresponds to a walk in the three-dimensional muqarnas from the lowest tier to the upper most tier. This lowest tier is found in the muqarnas graphs by arrows incident to bottom boundary nodes of the plan. The largest distance of a bottom boundary node to the center can be found at the diagonal of the plan. Therefore, most probably the longest path in the muqarnas graph follows a path approximating the diagonal of the plan. By putting islands on the diagonal these paths are interrupted and the maximal length of the paths decreases which results in muqarnas reconstructions containing less tiers: the bottom is “nearer” to the center. Therefore, we remove a diagonal node and consider the heights of the different possible muqarnas reconstructions. We will see in Chapter 5 that by trying to remove nodes on the diagonal, we are indeed often able to locate the islands in Il-Khanid muqarnas.

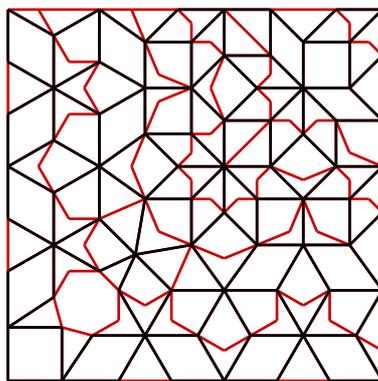


Figure 3.23. Plan of portal to enclosure at Sultan Han near Kayseri.

Removing the right edges in a plan not only consisting of basic elements, as in Seljuk muqarnas appear, can be more complicated. For the muqarnas only consisting of basic elements, most edges we need to remove are recognized as they have size unequal to the module. In Seljuk muqarnas the curved sides can have arbitrarily size and therefore different measures of edges in the plan may appear. In Figure 3.23 we see an example of a plan corresponding to a Seljuk muqarnas. The edges not corresponding to curved sides are red marked. The black edges, corresponding to curved sides do not all have the same length, the polygons near to the bottom boundary have larger sizes as the polygons in the center. To recognize the edges which correspond to back sides of elements standing on front sides of intermediate elements, we remark that these can be recognized as they divide a polygon of four edges, e.g. a parallelogram or a square, into two triangles or into two polygons of again four edges.

In the examples in Chapter 5 we will see that we are often able to remove most edges not corresponding to curved sides automatically by using the remarks before. Some edges still need to be removed by hand, but these edges are mostly easy recognized as edges not corresponding to curved sides as they have uncommon length. To locate the

islands, is still a process of trial and error. Our software is able to do suggestions, which diagonal nodes to remove, but cannot decide by its own.

3.5 Uniqueness of the Reconstructions

During the reconstruction process, the muqarnas plan is converted into a directed subgraph and this subgraph is extended to a three-dimensional muqarnas structure. In both steps, there may be some freedom which lead to different muqarnas reconstructions. In the first part, the conversion from muqarnas plan to a directed subgraph, these differences can be caused by removing nodes in the plan (see Section 3.4) or by changing the direction of undetermined edges (see Section 3.2). In the second part, the extension from the directed graph to a three-dimensional muqarnas structure, sometimes polygons can be interpreted as different element combinations (see Section 3.3.4). The answer to our question, whether muqarnas are uniquely reconstructible, will therefore be negative. In this section we study how muqarnas having the same plane projection can differ. Examples can be found in Chapter 5.

At first, we consider the situation that removing nodes on the diagonal results in different muqarnas reconstructions. By removing these nodes, an island in the plan emerges and a certain element combination on the diagonal is forced in the muqarnas. If we fix locally the elements, we need to ensure that the adjacent elements still fit to this fixed part. This is in general not the case, and we need to change more elements to create an approved muqarnas structure. Therefore, after removing nodes not only the elements on the diagonal vary. We have seen in Section 3.4 that this can influence the number of tiers in the muqarnas. Because more elements are involved and the shape of the muqarnas is changed, we call these muqarnas being *globally* different. These variations need to be considered in the algorithm by finding the subplan from the complete plan.

Secondly, while converting the subplan into a subgraph some directions of orbits may be undetermined. Giving these orbits different directions results in different muqarnas reconstructions. In this case, also more elements are involved and the different reconstructions can have different shapes consisting of a distinct amount of tiers. Therefore, we say that the different possible directions for the orbits which are undetermined by the constructing of the subgraph from the subplan, lead also to globally different computer reconstructions.

In Section 3.3.4 we have seen some examples of *locally* different muqarnas. Here only a few elements are exchanged, without changing the shape of the muqarnas. We have discussed the situation where a polygon in the plan is interpreted as the plane projection of a cell or of a combination of a cell together with an intermediate element. If the polygon was interpreted as the plane projection of a cell only, the back sides of this cell stand on the curved sides of an element in the previous tier. We can remove the cell and fill the emerged gap with a cell and an intermediate element. The intermediate element appears in the previous tier. The adjacent elements of the removed cell are not

changed. If we reconstruct the muqarnas from a simplified plan, we need to decide about splitting elements during the reconstruction from the subgraph to the muqarnas vault. The muqarnas graphs of these muqarnas vaults are the same.

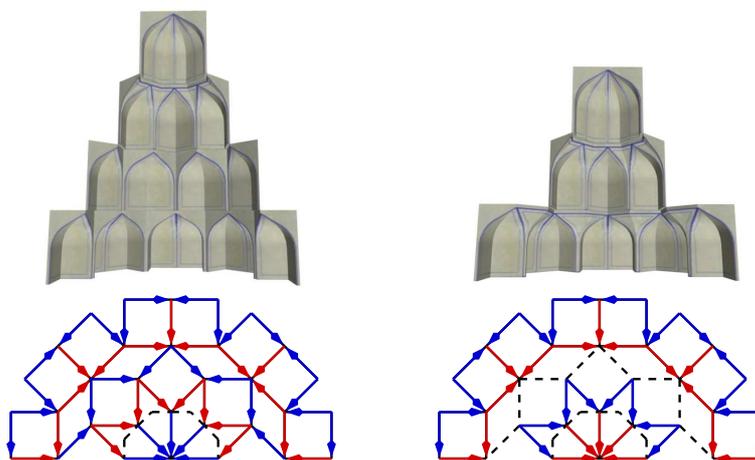


Figure 3.24. The cells in the second tier of the left reconstruction are pushed down in form of intermediate elements in the right picture.

In a similar way we can construct locally different muqarnas if it is possible to push down a complete tier. If a tier only consists of cells, with all backsides of the cells standing on curved sides of muqarnas elements, we can exchange all these cells by intermediate elements and push them to the previous tier. All the upper tiers then also need to be pushed down one tier. In Figure 3.24 an illustration of this situation is given. The second tier of the left reconstruction only consists of cells with plane projection a rhombus. In the right reconstruction these rhombi are interpreted as intermediate elements appearing in the first tier. The right reconstruction has then one tier less as the left reconstruction. If we look at the corresponding muqarnas graph of the right situation, we see that we need to remove the dotted edges from the plan before calculating the graph. The direction of the remaining edges does not change. Our algorithm is not able to reconstruct this right muqarnas correct from its muqarnas graph as this graph is disconnected. The Algorithm A-3.3 for calculating the height of the nodes only works for connected graphs. We therefore only reconstruct the left situation. In our muqarnas reconstruction we need to be aware that maybe tiers only consisting of cells with their backsides standing on curved sides need to be removed and intermediate elements in the previous tier inserted. We call this a local difference, although the amount of tiers changes. This is because we only exchange the elements of one tier, in such a way that the new elements still fit in the muqarnas. Adaption of adjacent elements is not required.

Finally, we want to mention the situation of half cells at the bottom of the muqarnas. If place for such half cells is available, we can insert them, but also leave them out. The designer needs to declare whether a half cell or a gap is preferred. It is decision which leads to locally different muqarnas.

Chapter 4

Software Tools for Reconstructing Muqarnas

In this chapter we describe three computer programs, which are developed to reconstruct muqarnas. The first program, `plantograph`, is written for converting a muqarnas subplan into a muqarnas graph (see Section 4.1). This subplan consists only of the edges which appear in the muqarnas graph. The task of `plantograph` is to give the edges a direction. A second program, `removelines`, is used to find the subplan from the complete muqarnas plan (see Section 4.2). The third program, `graphtomuq`, calculates a three-dimensional muqarnas reconstruction from the muqarnas graph with the initial plan as plane projection (see Section 4.3). For this aim, a list of muqarnas elements is calculated, together with a list containing the positions of the elements.

For the visualization of the two-dimensional objects, that are the plans and graphs, we use the *xfig*-file format, which is designed for the open source program *xfig*. The three-dimensional data are given in the VRML 2.0-file format. The *xfig*-files are then used as input and the computer reconstructions are given in VRML 2.0. As the calculations are all done in our own data-structures, the muqarnas reconstruction algorithm does not depend of these input and output formats. A short description of the *xfig*-file format can be found in Section 4.1.2 and the VRML 2.0-format is explained in Section 4.3.2.

4.1 The Program `plantograph`

To convert a muqarnas plan into a muqarnas graph we first determine the *subplan* consisting of the plane projection of all the curved sides of the elements and of backsides of cells appearing at the bottom of the muqarnas. We then give each edge in this subplan a direction to set the muqarnas graph (see Section 3.2). The determination of the directions is executed by the program `plantograph`.

In this section we first describe the program flow of the program `plantograph` (see Section 4.1.1). After that, we work out some parts more in detail. In Section 4.1.2, the

data-structures used for the plans and graphs in the program is explained. After that, we explain in Section 4.1.3 how we can convert the input to our intern data-structure.

To set the direction of the edges we use the rules which we formulated in Section 3.2. The first two rules depends on the boundary of the plan. In Section 4.1.4 we explain how this boundary is calculated. The main process of the program, the execution of the rules, is then given in Section 4.1.5. Finally, in Section 4.1.6, some remarks about the computing time of the program are given.

4.1.1 Conversion of the Plan into the Graph

The program `plantograph` applies the rules as presented in Section 3.2 to set as many directions of the edges as possible. With the additional tool `planedit`, written by Daniel Jungblut (see [Jungblut, 2005]), it is possible to set the direction of the remaining edges manually.

The program `plantograph` uses the complete muqarnas plan P and the subplan Q , which consists of the edges which appear in the muqarnas graph G . The following command line options are available to call the program `plantograph`:

- Q** <string> This option is necessary. It is followed by the name of the file representing the input muqarnas subplan Q .
- P** <string> The option **-P** should be followed by the name of the file representing the complete muqarnas plan P . If this option is not given, the subplan Q is used instead.
- G** <string> This option is followed by the name of the output file representing the muqarnas graph G . By default, we use the name of the input file, representing the subplan Q , with the prefix `graph_` placed before.
- p** <integer> The option **-p** can be followed by the integers 1, 2 or 4, representing a full, half or quarter muqarnas plan, respectively. By default, the input plan is considered being a quarter plan.
- c** If this option is given, an irregular center is allowed in the reconstruction.
- r** With this option, we work with a data-structure where nodes in the plan are stored as floating-point numbers. Without this option, we work with integers to represent the nodes in the graphs.

The task of the program `plantograph` is to set the direction of the edges of the subplan Q . Besides of the subplan Q , the complete plan P is required in the input. We recall from Section 3.2 that for giving the edges a direction, opposite edges are bound in orbits. For each orbit a direction is given. To decide whether edges are opposite we

need the information of the subplan Q as well as the information of the complete plan P . For setting the orbits the complete plan is only scanned to check whether a polygon has a cross edge connecting the edges which are candidates for being opposite (see second picture in Figure 3.5). The edges of a polygon with cross edge are not considered in our definition of opposite, because they could correspond to the projection of half intermediate elements joining at their front. In that case the edges which are candidates for being opposite do not necessarily have the same direction. As this situation rarely appears, in most cases the muqarnas graph can still be calculated correctly without information from the complete plan P .

In Figure 4.1 the program flow of the program `plantograph` is visualized. The thick arrows designate the main stream of the program. Dotted arrows are used to indicate that the objects are not changed, but serve as information. The first step in our algorithm is to

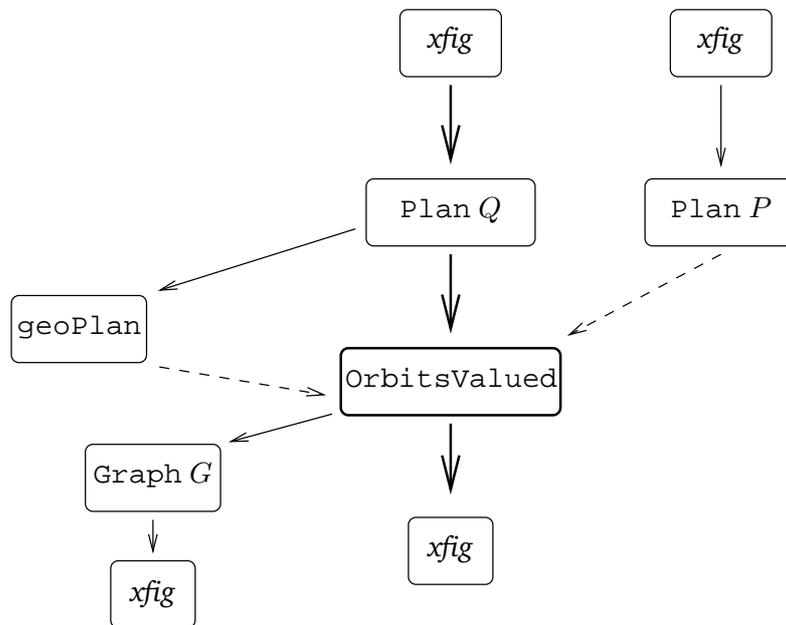


Figure 4.1. Program flow of the program `plantograph`.

read the input plan and convert it into our own data-structure, represented by the object `Plan`. In this way we create the subplan Q and the plan P containing the coordinates of the nodes in the muqarnas plan and their connections. Besides the object `Plan`, we create an object called `geoPlan`. The object `Plan` is responsible for the structure of the plan which is given by the connections between nodes. The object `geoPlan` cares about the features of the plan depending on the coordinates of the nodes. It consists of the boundary information of the plan and stores which section of the muqarnas is represented by the plan given in the input. This is given by the number behind the command line option `-p`.

As the direction of parallel edges is the same, we first analyze which edges are parallel. We store parallel edges in an orbit, which is represented by a vector of edges. In

this way we create a vector of the orbits with for each orbit a value to fix the direction of the orbit. This vector of orbits together with a value for its direction is stored as an object `OrbitsValued`. In this object our algorithm steps, as presented in Section 3.2 are applied to set the directions of the orbits. We then can directly convert this object into an *xfig*-file where edges with unknown directions are drawn by dotted lines. If the direction is known we use arrows to represent these directions.

In the remaining part of this section we give a more detailed description of the used data-structures and explain the steps executed during the program flow.

4.1.2 Representation of the Muqarnas Plan

The open source program *xfig*, written by Supoj Sustainthavibul and improved by Brian V. Smith and Paul King, is a program for drawing pictures in vector format. The *xfig*-file format is well documented in the *xfig*-manual (see [Sato and Smith, 2002]). We use this file format for the input plans and the output graphs. This makes it possible to visualize the graphs.

In the *xfig*-file format different drawing objects, like polygons, splines or circles are defined. Most objects are given by two or more lines containing numbers (see Figure 4.3). The first line starts with a number which is a code for the kind of object defined. For example, a line starting with the number 2 defines a polygon. The remaining numbers of this first line decodes some properties of the object. These properties can for example consist of information about the line thickness or color of the object. The second line consists of coordinates of nodes defining the object. In a possible third line extra properties of the object are defined. For example, in the case of a polygon drawn with arrows, these extra properties contain information about the drawing style of the arrows.

To represent muqarnas plans and graphs in the *xfig*-format only the object polygon is of interest for us. An open polygon in the *xfig*-format is represented by a sequence of coordinates of the nodes connected to each other. These coordinates are given in the second line of the object polygon. The first line contains, beside of the information that we are dealing with a polygon, also information whether the nodes are connected by arrows or by lines. In case of arrows this line also contains their direction. In this way a plan or graph is in the *xfig*-file format represented by a list of edges, where each edge is given by an open polygon defined by the two connected nodes. In the case of a plan the connections are given by lines and in case of a graph they are given by arrows.

In our software, the structure of a muqarnas plan and a muqarnas graph, are represented by an adjacency-list: each node in a plan is linked to a list of nodes representing its neighbors. A detailed description of this data-structure can be found in [Sedgewick, 2002, pp. 31-35]. In the case of a graph only the successors of a node are listed. In Figure 4.2 we see examples of such adjacency-lists corresponding to a plan and a graph. In the adjacency-list we refer to the nodes by using *indices*. These indices consist of a set of non-negative integers so that each node uniquely corresponds to an index taken from this set. For a plan $P(N, E)$ we denote with $I_P \subset \mathbb{Z}_{\geq 0}$ the set of index

numbers of the nodes N .

If the command line option `-r` is not given, we store the coordinates of the nodes as integers. This is possible if all nodes in the plan $P(N, E)$ are in the grid $\mathbb{L} := (\mathbb{Z} + \mathbb{Z}/\sqrt{2}) \times (\mathbb{Z} + \mathbb{Z}/\sqrt{2})$. We have seen in Section 2.2.3 that if a node $v \in \mathbb{R}^2$ can be represented by integers $a, b, c, d \in \mathbb{Z}_{\geq -1}$ so that $v = (a + b/\sqrt{2}, c + d/\sqrt{2})$ then we can represent it by one integer

$$\iota(v) := f(f(a + 1, b + 1), f(c + 1, d + 1))$$

with f as given in Equation (2.2). We use these integers given by the grid as our index numbers to refer to a certain node in our adjacency-list. The integer then directly includes the information of the coordinates of the elements.

In the case the command line option `-r` is given by calling the program `plantograph`, the coordinates are stored as floating-point numbers. We store additionally a list of the floating-point numbers. In this list each floating-point number is linked to an integer, which is the index of then node.

Let $v_P(i) \in \mathbb{R}^2$ the coordinates of the node corresponding to the index $i \in I_P$. We denote $i_P(v)$ for the index of the node $v \in N$ in the adjacent-list representation of the plan P . If we represent nodes v with integers $\iota(v)$ and use these integers for the indices, then for the subplan Q of P , $I_Q \subset I_P$ and $v_Q(i) = v_P(i)$ for $i \in I_Q$ as the coordinates of the nodes are determined by there integer representation. If we represent the coordinates with floating-point numbers this is not the case. The same index number in different plans in general refers to different nodes, that means for $i \in I_Q \cap I_P$ in general $v_Q(i) \neq v_P(i)$, they depend of the order the nodes are inserted in the node-list. This means that if we work parallel with different copies of a plan and switch between them, we have to look up the coordinates corresponding to a certain index and compare the coordinates directly instead of comparing the indices.

We have chosen to use an adjacency-list as data-structure for our graphs so that we have a fast access to neighboring nodes. For the directed graphs this means that we have only fast access to the successors of a certain node. Therefore, we often work parallel with the reverse graph G_{rev} of G . Graph G_{rev} has the same set of nodes and edges as the graph G but the edges are directed in the reverse order compared to G . The graph G_{rev} gives then the possibility to have also fast access to the predecessor of a node in G .

4.1.3 Conversion of the Input into a Muqarnas Plan

The first task of the program `plantograph` is to convert a *xfig*-file into our intern data-structure of a plan P . To do this we start with an empty plan P , read the information of the *xfig*-file line by line and put this information in the plan P . While scanning the *xfig*-file, a polygon can be recognized by the first line of the object starting with the number 2 (see Figure 4.3). This first line also contains the informations whether this polygon is drawn with lines or with arrows. In the context of reading the plan we only consider the lines and if we read a graph from a *xfig*-file, we only store the arrow information. We scan the second line for the coordinates of the nodes connected to each other. The

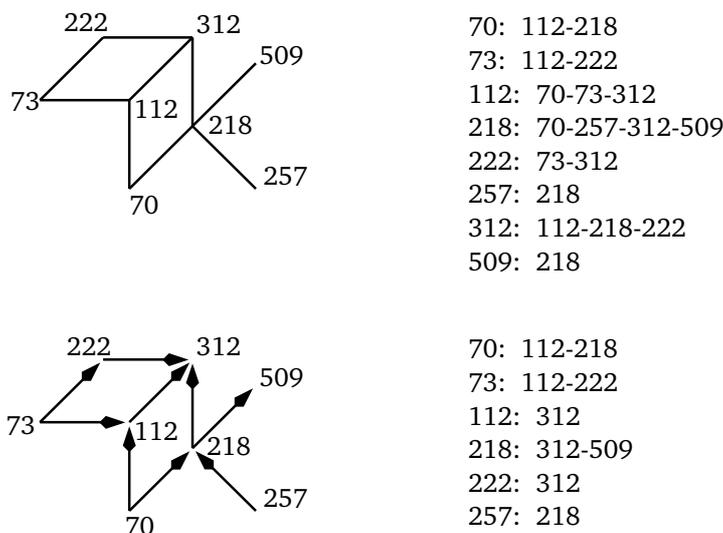


Figure 4.2. On the left: visualization of a plan and graph. On the right: connections between the nodes stored in an adjacency-list.

```

2 1 0 4 0 7 50 -1 -1 0.000 0 0 -1 0 0 2
900 1536 1536 1800

```

Figure 4.3. Part of a *xfig*-file which defines a polygon (line 1) given by two connected nodes (line 2).

two lines illustrated in Figure 4.3 define an open polygon. In the second line we see that the polygon is given by the connection between the node $v = (900, 1536)$ and the node $(1536, 1800)$. In this example the unit is chosen to be equal to 900, so that we will refer to these coordinates with floating-point numbers $v = (1, 1.71)$, $w = (1.71, 2)$.

The method we use to convert the information of the *xfig*-file into our own data-structure depends on whether we represent our node with integers or with floating-point numbers. In the situation that we use an integer representation for the nodes $v = (v_x, v_y) \in \mathbb{R}^2$, we need to be able to find $a, b, c, d \in \mathbb{Z}_{\geq -1}$ such that $v_x \approx a + b/\sqrt{2}$, $v_y \approx c + d/\sqrt{2}$, then we can convert (a, b, c, d) into one integer by calculating

$$\iota(a + b/\sqrt{2}, c + d/\sqrt{2}) := f(f(a + 1, b + 1), f(c + 1, d + 1))$$

as defined in Equation (2.5) in Section 2.2.3.

Like in Section 2.2.3, we use again the notation $\lfloor r \rfloor$ to denote the largest integer smaller than or equal to r . To write a given floating-point number r by integers (a, b) such that $r = a + b/\sqrt{2}$, we need to find a, b such that

$$b \approx \sqrt{2}(r - a).$$

There does not necessarily exist an optimal solution such that the error, which is given by $|\lfloor \sqrt{2}(r - a) + \frac{1}{2} \rfloor - \sqrt{2}(r - a)|$, is minimal. We choose a of a finite set $\mathbb{Z}_{[-k,k]} := \{-k, -k + 1, \dots, k - 1, k\}$, for a certain k , such that

$$|\lfloor \sqrt{2}(r - a) + \frac{1}{2} \rfloor - \sqrt{2}(r - a)| \quad (4.1)$$

is minimal for $a \in \mathbb{Z}_{[-k,k]}$. This means that for the a^* for which Equation (4.1) is minimal, the corresponding $b^* = \lfloor \sqrt{2}(r - a^*) + 1/2 \rfloor$.

The representation by integers is only possible if the muqarnas consists of basic muqarnas elements. We will use that in such a muqarnas plan there is always a certain distance between the nodes in the plan. If a node v appears in a plan and a node w appears in the plan such that $\|v - w\| \leq 1 + 1/\sqrt{2}$, then they need to be connected by edges belonging to a square or rhombus based figure (see Figure 4.4). We find that in that case

$$\begin{aligned} \|a_v - a_w\| &\leq 1, & \|c_v - c_w\| &\leq 1, \\ \|b_v - b_w\| &\leq 2, & \|d_v - d_w\| &\leq 2, \end{aligned} \quad (4.2)$$

hence $a \in \{-1, 0, 1\}, b \in \{-2, -1, 0, 1, 2\}$. This means that if we know the integer

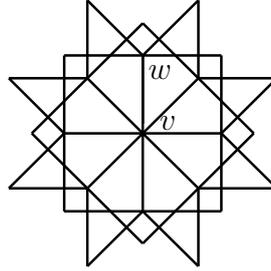


Figure 4.4. Adjacent nodes in a muqarnas plan consisting only of basic elements are connected by an edge belonging to a rhombus or a square based figure.

representation (a_v, b_v, c_v, d_v) of a node v , we can calculate the integer representation of neighboring nodes w by setting $r = v_x - w_x$ and calculating the minimum of Equation (4.1) with $a \in \{-1, 0, 1\}$. Then in the integer representation (a_w, b_w, c_w, d_w) of w the a_w is given by $a_w = a_v - a$. Analogously we can calculate c_w by setting $r = v_y - w_y$ in Equation (4.1). In general, if $\|v - w\| = l$ we choose $a := a_v - a_w$ from the set $\mathbb{Z}_{[\lfloor -(l+1) \rfloor, \lfloor l+1 \rfloor]}$.

We set the integer representation of the nodes by comparing their coordinates with the coordinates of the nodes in their neighborhoods of which the integer representation is already known. To apply this idea, we first read all coordinates of the *xfig*-file and list them as a vector of edges, where edges are given by two nodes with floating-point coordinates. This vector is the set of floating-point numbers we need to convert. As we would like to have $a, b, c, d \geq -1$ so that we can calculate $\iota(a + b/\sqrt{2}, c + d/\sqrt{2})$, we first translate the plan so that all coordinates are positive. We search the corner

(x^*, y^*) such that $v_x \geq x^*$, $v_y \geq y^*$ for all nodes $v \in N$ in the muqarnas plan $P(N, E)$. By subtracting (x^*, y^*) from all coordinates v , all coordinates can be represented with positive coordinates and (x^*, y^*) is represented with integers $(a, b, c, d) = (0, 0, 0, 0)$.

The nodes get all a status 0. which has the meaning that the integer representation of the node is not known. A node v will get status 1 if its integer representation is known, but not of all its neighboring nodes the integer representation is known. If for a node both its integer representation is known as well as the integer representation of its neighboring nodes, then it gets status 2.

We start the converting process by giving the origin status 1. For each node v with status 1 we look for w with status 0 such that $\|v - w\| < 1 + 1/\sqrt{2}$ and determine its integer representation. The node w gets status 1 and if for all w with $\|v - w\| < 1 + 1/\sqrt{2}$ the integer representation is determined, the node v gets status 2. If the set of nodes with status 1 is empty, but there are still nodes v with status 0, we have to look for the nearest node w of v with status 2 and estimate its integer representation by choosing $a = a_v - a_w$ from the set $\mathbb{Z}_{[\lfloor -(\|v-w\|+1) \rfloor, \lfloor \|v-w\|+1 \rfloor]}$. The converting process will stop if there are no nodes of status 0 anymore.

After converting all floating-point numbers to integers, the edge-list is used to link each $v \in N$ to its neighbors w by linking $\iota(w)$ to $\iota(v)$. The object `Plan` contains a method `insert($\iota(v), \iota(w)$)` which task it is to add $\iota(w)$ to the neighbor list of $\iota(v)$.

If we work with a data-structure based on floating-point numbers, we can directly add the edges which we find in the `xfig`-file to the plan P . The object `Plan` contains a method `insert(v, w)`, for nodes $v, w \in \mathbb{R}^2$ given by its coordinates read from the `xfig`-file divided by the unit u . This method first observes the actual node-list of the plan P to check whether their are already indices $i_P(v)$ and $i_P(w)$ given. If they are not available in the node-list, we create index-numbers $i_P(v)$ respectively $i_P(w)$ not existing yet in I_P . The connection $i_P(v), i_P(w)$ is then added to the linked list by adding $i_P(v)$ to the neighbors of $i_P(w)$ and adding $i_P(w)$ to the neighbors of $i_P(v)$.

4.1.4 Boundary of a Muqarnas Plan

From the subplan $Q(M, C)$, given in the input, we create the object `GeoPlan`, which main task it is to store the boundary information of the plan. This object works with plans corresponding to a muqarnas in a niche or dome. For reconstructing more general muqarnas forms, the methods in this object need to be adapted. It uses the information which section of the muqarnas is represented in the plan, as given by the command line option `-p` in calling `plantograph`. It stores the number given behind `-p` in the command line and uses it to calculate the boundary, center and bottom boundary nodes of the plan. They are stored as vectors of index numbers referring to the corresponding nodes. We restrict to describe the situation dealing with a quarter muqarnas plan in the input, as we also gave the boundary definitions in that context. If a larger part of the muqarnas plan is given, we can use symmetry arguments to calculate the complete boundary.

The *boundary nodes* are determined directly from Definition 2.2.8. We iterate over

all $v \in M$ to decide for each node whether it belongs to the boundary or not. For this, we select for each node $v \in M$ its neighbors $\{a_0, \dots, a_{n-1}\}$ and order them such that $\alpha(v, a_0, a_i) < \alpha(v, a_0, a_{i+1})$. We apply the definitions of *figure* (see Definition 2.2.6) and *island* (see Definition 2.2.7) to decide for all $i = 0, \dots, n-1$, $j = (i+1) \bmod n$ whether $\{v, a_i, a_j\}$ defines a figure or an island. The definition of the boundary shows that v is in the boundary if there are i, j with $i = 0, \dots, n-1$, $j = (i+1) \bmod n$ such that $\{v, a_i, a_j\}$ neither defines a figure nor an island.

To determine the *center nodes* (see Definition 2.2.9), we select the nodes on the boundary between the end-node

$$v^* = \left(\max_{\{v \in M: v_y = y_{\max}\}} v_x, y_{\max} \right)$$

of the x -mirrorline and the end-node

$$w^* = \left(x_{\max}, \max_{\{w \in M: w_x = x_{\max}\}} v_y \right)$$

of the y -mirrorline. The nodes v^*, w^* can be found by comparing the coordinates of all nodes in the boundary and select from the nodes with maximal x -value the one with largest y -value and similar from the nodes with maximal y -value the node with largest x -value is selected. The shortest path between v^* and w^* is set with assistance of a multiplication operator \circ defined on plans. We define for a plans $P(N, E)$ and $Q(N', E')$ the product $[P \circ Q] (\tilde{N}, \tilde{E})$ by

$$\text{if } \text{edge}(u, v) \in E \text{ and } \text{edge}(v, w) \in E' \Rightarrow \text{edge}(u, w) \in \tilde{E}.$$

The set of nodes $\tilde{N} \subset \{N \cup N'\}$ is given by the nodes incident to the edges $e \in \tilde{E}$. We use the notation

$$P^k := \underbrace{P \circ \dots \circ P}_{k \text{ times}}$$

for applying k times the operator \circ . For a plan P there is a path between v and w of length k if $\text{edge}(v, w) \in P^k$.

For computing the shortest path between v^* and w^* , which defines the center, we store the subplan given by the boundary nodes of the plan as a subplan R . We determine for $i = 0, \dots, k$ the plans R^i where k is chosen so that this is the smallest integer for which $\text{edge}(v^*, w^*) \in R^k$. We construct the path of length k defining the center by finding a sequence of nodes s_i such that

$$\text{edge}(v^*, s_i) \in R^i \text{ and } \text{edge}(s_i, w^*) \in R^{k-i} \text{ for } i = 1, \dots, k-1.$$

The process of calculating the center is illustrated in Figure 4.5. Here, the plans R^i are drawn for the plan corresponding to the muqarnas in the basement of the north iwan of the Friday Mosque in Natanz. We see that $\text{edge}(v^*, w^*) \in R^6$, so the center is given by a

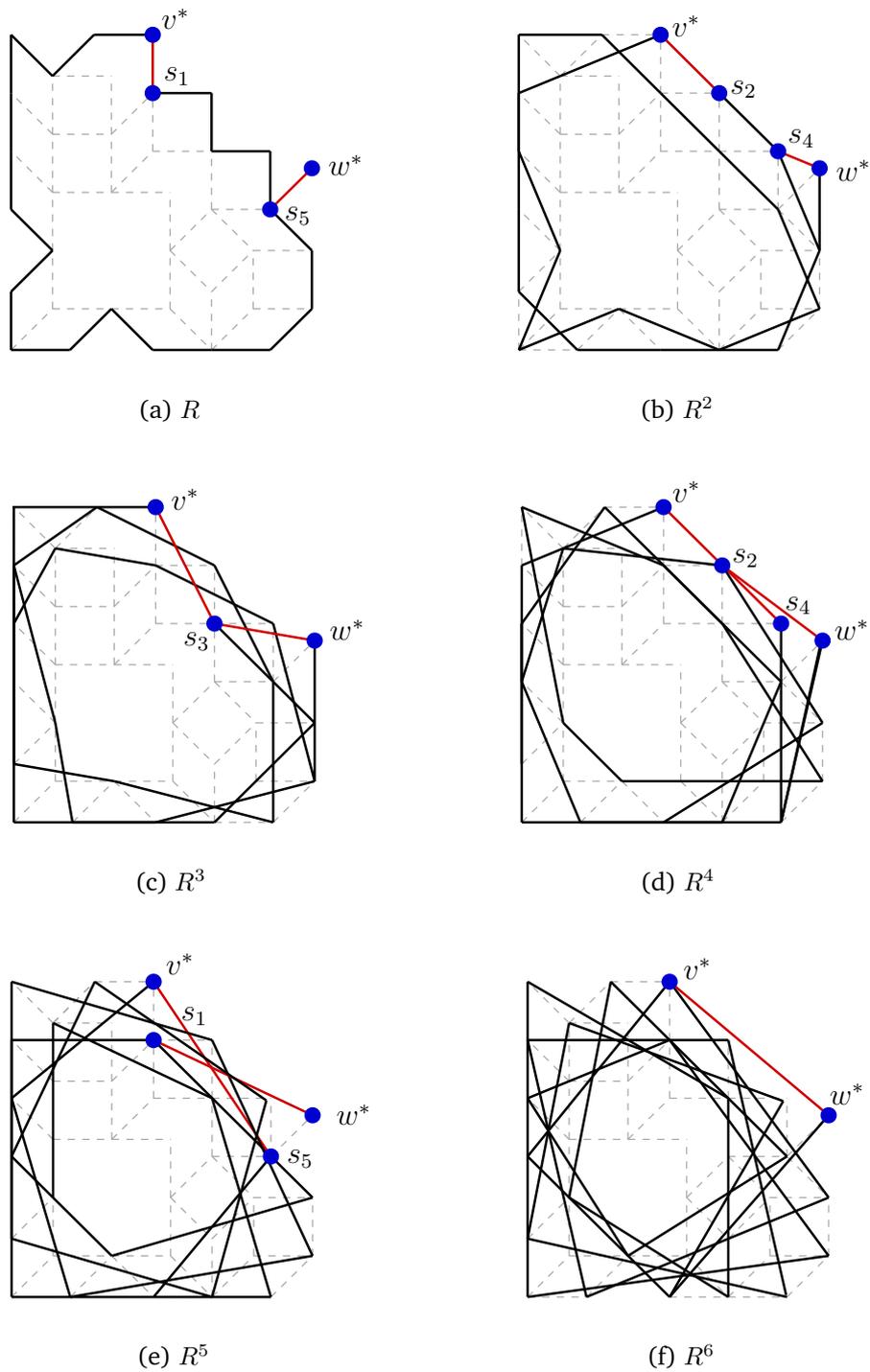


Figure 4.5. The center of the plan corresponding to the basement of the north iwan of the Friday Mosque in Natanz is calculated by multiplication of the subplan defined by the boundary nodes.

path of length 6. It is defined by the sequence consisting of the common neighbors s_i of v^* in R^i and of w^* in R^{6-i} .

If the shortest path between nodes v and w is long, then k is large and a lot of space is needed to store the plans R^i for $i = 1, \dots, k$. This means that the above described method for calculating the shortest path between two nodes needs a lot of computer capacity. To calculate the center of a muqarnas plan, this is not a problem, as the center is mostly small. The path defining the *bottom boundary* (see Definition 2.2.10) can be larger. Therefore, we have chosen to use a different method in that case.

From the set of boundary nodes we delete the center nodes and the nodes on the x -mirrorline and the y -mirrorline. From the remaining set of nodes L , we calculate the maximal subplan $R(\tilde{L}, D)$ of $Q(M, C)$ given by these nodes. That means, that if $\text{edge}(v, w) \in C$ and both $v, w \in L$, then $v, w \in \tilde{L}$ and $\text{edge}(v, w) \in D$. We add the nodes from the x -mirrorline and the y -mirrorline with minimal x and minimal y -value respectively.

4.1.5 Determination of the Direction of the Edges

The main object during the program flow is the object `OrbitsValued`. In this object the orbits of the muqarnas plan are stored and we apply the rules as formulated in Section 3.2 on this object to set the directions of the orbits. After setting the direction, we return the information of this object as a graph.

An edge $\text{edge}(v, w)$ is stored as set of two nodes v, w . In this context we do not identify $\text{edge}(v, w)$ with $\text{edge}(w, v)$. An orbit is a vector of edges and the object `OrbitsValued` contains a vector of orbits Ω along with for each orbit Ω an integer $r(\Omega) \in \{-1, 0, 1\}$. If $r(\Omega) = -1$, the edges in the orbit Ω point from w to v . The value $r(\Omega) = 0$ means that the direction of the edges in Ω are undetermined. If the edges in the orbit Ω point from v to w then $r(\Omega) = 1$.

Let $P(N, E)$ be the complete muqarnas plan and $Q(M, C)$ the subplan of which we need to determine the direction of the edges. To set the orbits in Q according to P , we work with a copy D of the set of edges C from the subplan $Q(M, C)$. We calculate the orbit Ω to which an edge $e \in D$ belongs with the definition of parallel edges (see Definition 3.2.1). Let $\Omega_e(C)$ be the orbit in C to which the edge e belongs. To find the orbit $\Omega_e(C)$ is done by using a recursive process split in three steps:

- (i). Insert e to the orbit $\Omega_e(C)$.
- (ii). Delete e from D .
- (iii). If there is an opposite edge f of e , then insert $\Omega_f(D)$ to the orbit $\Omega_e(C)$.
Else stop and return the orbit $\Omega_e(C)$.

We insert for the edge e its orbit $\Omega_e(C)$ to the object `OrbitsValued`, its direction is marked by $r(\Omega_e(C)) = 0$. We set the next orbit by choosing an arbitrarily edge f of the actual set of remaining edges D and insert $\Omega_f(C)$ in the object `OrbitsValued` with

$r(\Omega_f(C)) = 0$. This process repeats till the set D is empty, in that case all orbits are set. After setting all orbits, $r(\Omega) = 0$ for all orbits Ω .

After all preparation work, that consisted of calculating the boundary of the plan and setting all orbits, we apply our rules on the object `OrbitsValued` to set the directions of the orbits.

For applying Rule 1, arrows point from non-singular bottom boundary nodes inward, we use the bottom boundary information from the object `geoPlan`. First, the vector of bottom boundary nodes need to be filtered, as we can only apply Rule 1 on the non-singular bottom boundary nodes. Non-singular bottom boundary nodes are defined in terms of their neighbor nodes. We therefore convert the vector of bottom boundary nodes into a path, so that we have fast access to the neighbors u, w of a node v . The path representing the bottom boundary is stored as a plan $R(W_{bb}(Q), \tilde{E})$, with $W_{bb}(Q)$ the set of bottom boundary nodes in Q and $\tilde{E} \subset E$ such that $e \in \tilde{E}$ if its end-nodes $e_s, e_t \in W_{bb}(Q)$. Then for each $v \in W_{bb}(Q)$ we can calculate whether v is a singular or non-singular bottom boundary node (see Definition 3.2.3) by calculating the line $l(u, w)$ and check if $l(u, w)$ separates v from the plan Q . The set of non-singular bottom boundary nodes is stored as a vector. For each non-singular bottom boundary node v we iterate over its neighbors in the plan Q . If we find a neighbor w which is not a non-singular bottom direction, then v points to w in the muqarnas graph. We iterate over all orbits Ω in the object `OrbitsValued` to find the orbit Ω containing $\text{edge}(v, w)$ or $\text{edge}(w, v)$. If Ω contains $\text{edge}(v, w)$ we set $r(\Omega) = 1$, otherwise if Ω contains $\text{edge}(w, v)$ we set $r(\Omega) = -1$.

To apply Rule 2, arrows with one end-node incident to the center points to the center, we work similar. In this case for each center node v we look for its adjacent nodes w not in the center. Then there is an orbit Ω with $\text{edge}(v, w) \in \Omega$ or $\text{edge}(w, v) \in \Omega$ and the direction of this orbit is set so that w points to v . If the command line option `-c` is given by calling to `plantograph`, an irregular center is allowed and we need to apply the alternative Rule 2. Before setting the direction of the orbit Ω to which $\text{edge}(v, w)$ or $\text{edge}(w, v)$ belongs, we need to check if this orbit does not contain an edge $\text{edge}(v', w')$ with both $v', w' \in W_c(P)$. If such an edge exist the direction of Ω cannot be set.

For applying Rule 3, set the direction of the islands, we iterate over the orbits Ω for which the direction is unknown: $r(\Omega) = 0$. All the edges of an undetermined orbit are tested on being a member of an island. To test whether an edge $\text{edge}(v, w)$ belongs to an island is done by iterating over the neighbors k of v . We check for each neighbor the sets $\{v, k, w\}$ and $\{v, w, k\}$ for being an island. If an island is found it is stored as a sequence of nodes v_0, \dots, v_n . The next step is then to determine the direction of the edges $\text{edge}(v_i, v_{i+1}), i = 0, \dots, n - 1$. For this we find the node v_r for a $r \in \{0, \dots, n\}$ with largest x -coordinate by comparing the coordinates of the nodes v_0, \dots, v_n . The island is stored as a graph with arrows as given by Rule 3. Finally we bring over the directions of these arrows to the to the orbits in the plan. For an arrow $\text{arw}(v, w)$ in the island we look for the orbit in `OrbitsValued` containing the corresponding edge $\text{edge}(v, w)$ or

$\text{edge}(w, v)$. The direction of the corresponding orbit can then be set.

For applying Rule 4, the graph has no niches, we iterate over the orbits. For the orbits of which the direction is undetermined the direction is set arbitrarily. If this result in a muqarnas graph containing a niche, we know that the direction should be swapped. For both possible directions we execute this test, if both resulting graphs have no niches the direction of the orbit is set back to unknown.

Finally, the contents of the object `OrbitsValued` is written in a *xfig*-file. For each orbit we first a comment line in the *xfig*-file is written to denounce the begin of a new orbit. This is done so that the orbit information is not lost. After that, all edges of the orbit are inserted as arrows according to its direction. If the direction is not known we insert an *xfig*-object representing a dotted line. The program `planedit` (see [Jungblut, 2005]) is written for interactively changing the graph. This program offers the possibility to change the directions of the orbits manually. The directions of the undetermined orbits can also be set.

4.1.6 Complexity Analysis

We finish this section with some remarks regarding the computing time of the methods applied during the runtime of the program *plantograph*. We want to emphasize that the muqarnas plans are quite small and in practice computing time is not main point. We favor a clearly implementation over finding the most efficient algorithm. We restrict ourselves by presenting raw estimations showing that the algorithms are of polynomial time. The computing time is formulated according to the *size* of a muqarnas plan. The size of a muqarnas plan is given as the number of nodes $|N|$ in the plan. We announce the computing time by using Landau's symbol \mathcal{O} .

In a muqarnas plan a node can only have a few neighbors, independently of the total amount of nodes in the plan. Looking at the smallest angle in a muqarnas plan consisting of basic elements only, we find that this is the angle of a rhombus and equals to $\pi/4$. A node in such plan can thus have maximal eight neighbors, independently of the size of the plan. For more general muqarnas, where a larger variation of angles is allowed, the number of neighbors of a certain node is also restricted and does not depend on the amount of nodes in the muqarnas plan. From this we can conclude that the number of edges in a plan depends linearly on the number of nodes in the plan. For a muqarnas consisting of basic elements, for example, we can state that

$$|E| \leq \frac{8|N|}{2} \quad (4.3)$$

as each node as at most eight neighbors. Because of this relation, we can always give the computing time in terms of $|N|$.

By reading a muqarnas plan $P(N, E)$ the *xfig*-file is scanned and every connection in the plan is inserted as an edge in our data-structure of the plan. Therefore, this scanning process needs computing time linearly dependent of the number of edges $|E|$. We find

that the computing time equals to $\mathcal{O}(|E|) = \mathcal{O}(|N|)$. To set the boundary of the plan, we iterate over the nodes N of the plan. We survey for each node $v \in N$ whether it is surrounded by figures or islands. The maximal size of an island does not depend on the size of the plan. The validation process, whether a given combinations of nodes defines a figure, does also not depend on the size of the plan. Therefore, checking whether a fixed node belongs to the boundary depends only on the neighbors of the node v . As the maximal number of neighboring nodes does not depend on $|N|$ we conclude that computing time for setting the boundary is given by $\mathcal{O}(|N|)$.

To set the center of a muqarnas graph, we calculate R^k , with $R(L, D)$ the path defined by the boundary nodes and k the number of center nodes. For calculating R^2 we look for each edge to neighbors if its end-point. We use this argument to conclude that calculation of R^k costs $\mathcal{O}(k \cdot |D|)$ operations. As $|D| < |E|$ and the center has always less than $|N|$ nodes, we can estimate computing time with $\mathcal{O}(|N| \cdot |E|) = \mathcal{O}(|N|^2)$.

For calculating the outer boundary, we iterate over the set of nodes on the boundary, and decide for each node whether it belongs to the center or a mirrorline. From the remaining nodes we calculate the maximal subplan defined by these nodes. This is also done by iterating over the nodes. We can verify that the computing time can be given by $\mathcal{O}(|N|)$.

In the recursive process, which we use for setting the orbits, for each edge we find parallel edges from the graph with the edges already used removed. That means that the i th edge, which we search, is taken from a set of $|E| - i$ edges. The number of operations is then not larger than

$$\mathcal{O} \left(\sum_{i=0}^{|E|} |E| - i \right) = \mathcal{O}(|E|^2) = \mathcal{O}(|N|^2).$$

Setting the direction of orbits according to Rule 1 is done by iteration over the nodes on the boundary. For each node v among the bottom boundary, we look for the orbit to which the edges joining in the node v belongs. The corresponding orbit is found by examining for each orbit its edges. In the worst case we iterate over all edges until we find the corresponding orbit. The number of boundary nodes is smaller than $|N|$ and the computing time of this rule can be estimated by $\mathcal{O}(|N| \cdot |E|) = \mathcal{O}(|N|^2)$. For Rule 2 we can use similar arguments to conclude that the computing time can be estimated by $\mathcal{O}(|N|^2)$.

During the execution of Rule 3, we iterate over the undetermined edges and validate whether these belong to an island. Setting the directions of the edges of an island is an operation which does not depend in the size of the plan, as the size of an island is bounded. For the edges in the island, we need to find the corresponding orbit by iterating over the edges in the orbit. we find that computing time can be given by $\mathcal{O}(|E|^2) = \mathcal{O}(|N|^2)$.

In Rule 4 we iterate over the orbits. The number of orbits can be maximal $|E|$. We give an orbit a direction and iterate over all nodes to check whether the resulting graph

contains a niche. The amount of operations depends then on $|E| \cdot |N|$ and computing time can be given by $\mathcal{O}(|E| \cdot |N|) = \mathcal{O}(|N|^2)$. Adding the computing times of the different methods executed in the program `plantograph`, we conclude that computing time of the program equals to $\mathcal{O}(|N|^2)$.

4.2 The Program `removelines`

The program `removelines` is written to do the first analyzes on the muqarnas plan. Its task is to calculate the subplan Q from a muqarnas plan P . This subplan Q consists of the edges which appear in the muqarnas graph G . The program `removelines` is not able to find all edges in the plan P , which not correspond to a curved side, but it is able to find most of them and it proposes suggestions to remove nodes to create islands in the plan. Creating islands can make it possible to create reconstructions containing less tiers. The user needs to decide whether these nodes indeed should be removed. In Section 4.2.1 we present the methods executed in the program. Some remarks about its computing time are given in Section 4.2.2.

4.2.1 Removal of Edges from the Plan

As input the program `removelines` requires a `xfig`-file representing the complete muqarnas plan. The output consists of a `xfig`-file with some edges removed and some text with suggestions of additional nodes to remove. Removing a node v means that we remove not only the node v , but also all edges incident to v . The following command line options are available to call the program `removelines`:

- P**<string> This necessary option is followed by the name of the file representing the complete muqarnas plan.
- o**<string> This option is followed by the name of the output file representing the subplan Q . Without this option, the name of the input plan with the prefix `rmlines_` placed before is used.
- c** With this option an irregular center is allowed in the reconstruction.
- h**<integer> This option is followed by a number representing how many tiers the corresponding muqarnas contains.
- r** With this option, we work with a data-structure where nodes in the plan are stored as floating-point numbers. Without this option, we work with integers to represent the nodes in the graphs.

If the option `-r` is not given, we assume that the muqarnas only consists of basic elements. We have seen in Section 3.4 that in that case we need to remove two kind of

edges. Those which correspond to backsides of cells standing on front sides of intermediate elements (back on front) and the edges corresponding to places where front sides of intermediate elements touch (front to front). Two methods are designed to handle these situations. The first method, `FilterLinesLengthOne()`, calculates for each edge in the input plan, not corresponding to the front border of the muqarnas, its length. If this length is unequal to one, the edge is removed from the plan.

The second method, `setHeight(unsigned int n)` is only executed if the command line option `-h` is given in the input. It uses the number n which follows after the option `-h` in the input line. The effect of removing nodes on the diagonal to create a muqarnas graph with maximal height n is analyzed by this method. For each diagonal node v the minimal height of a possible muqarnas reconstruction corresponding to the plan $P(N_{\setminus\{v\}}, E)$ is calculated. The notation $N_{\setminus\{v\}}$ is here used to denote the set N with the node v removed. The program `removelines` produces some text stating whether there is a reconstruction possible and what the minimal height of such a reconstruction is. Special attention is taken on the nodes which results in a reconstruction of height n .

The possibility of reconstruction and the height of the reconstruction is approximated as follows: we first calculate a graph corresponding to the plan $P(N_{\setminus\{v\}}, E)$ by applying the Rules 1–4 as formulated in Section 3.2. If the command line option `-c` is given the alternative Rule 2 is used. In the case that the rules are in a conflict situation, that means that different rules try to set the direction of the same orbit differently, there is no reconstruction possible. Otherwise, we calculate the height of the nodes in the subgraph $H(M', C')$ given by the edges for which the direction is determined. If there is a muqarnas reconstruction possible with muqarnas graph $G(M, C)$ then the graph H is a subgraph of this muqarnas graph G . This subgraph H does not need to be connected, as some edges of the muqarnas plan can be undetermined and therefore do not appear in the graph H . Albeit the graph H does not need to be connected, we approximate the height of the nodes H by applying Algorithm A-3.3. The algorithm calculates for each component in H the height of its nodes. If this calculation fails: that is it stops because $h_{max} > |M|$, there is a cycle in the graph and H cannot be a subgraph of a muqarnas graph. Otherwise, the Algorithm A-3.3 calculates for each component the height of the nodes well. If there is a reconstruction possible, these components are projections of smaller parts of the complete muqarnas. We need to add for each component a constant to the calculated heights, referring to the height of the smallest tier in which this muqarnas component appears. The calculated heights $\tilde{h}(v)$ of the nodes $v \in M'$ are then equal to or smaller than the height $h(v)$ of the nodes. We therefore can conclude that the number of tiers of the corresponding muqarnas is equal to or larger than the maximal height of the nodes in the subgraph and a lower boundary on the heights is determined.

If the command line option `-r` is given, more variations of elements can be used in the reconstruction. We have seen in Section 3.4 that the edges corresponding to the places where backsides of cells stand on front sides of intermediate element can often be recognized in the muqarnas plans as they divide a polygon of four edges in two polygons of four edges, or two polygons of three edges (see Figure 3.23). To handle the first case, the

method `DeleteDivisionLines()` is written. This method deletes all non-boundary nodes which have only two neighbors. The second case, where a polygon of four edges is divided in two triangles, is handled by the method `DeleteTriangleLines()`. It removes for each triangle the edge with most differing length. This is done because a triangle corresponds to a half element and the sizes of the curved sides of such a half element have in general the same length. Triangles are recognized in the plan as a node with two neighbors which are connected to each other.

4.2.2 Complexity Analysis

We finish this section with some remarks about the computing time of the methods used in the program `removelines`. We use again that the number of edges $|E|$ linearly depends on the number of nodes in a muqarnas plan (see Section 4.1.6). The method `FilterLinesLengthOne()` iterates over the edges in the plan to determinate for each edge its length. Therefore the computing time linearly depends on $|E|$, the number of edges in the muqarnas plan. Computing time is then given by $\mathcal{O}(|E|) = \mathcal{O}(|N|)$.

The method `setHeight(unsigned int n)` calculates for each diagonal node a possible muqarnas reconstruction. The muqarnas structure is calculated by applying `plantograph`. We have seen in Section 4.1.6 that this program needs $\mathcal{O}(|N|^2)$ operations. After the graphs are calculated, the Algorithm A-3.3 is used to calculate the height of the nodes. This Algorithm will call to the Algorithms A-3.1 and A-3.2 at most $|N|$ times. For executing both the Algorithm A-3.1 and A-3.2 we iterate at most $|N|$ times over the nodes in N . We find that the computing time of Algorithm A-3.3 can be given by $\mathcal{O}(|N|^3)$. Summarizing, iteration over diagonals, running `plantograph` and running Algorithm A-3.3 gives computing time of $\mathcal{O}(|N| \cdot (|N|^2 + |N|^3)) = \mathcal{O}(|N|^4)$.

For the methods `DeleteDivisionLines()` and `DeleteTriangleLines()` we remark that we iterate over the nodes in the plan to decide whether these nodes divide a polygon, or define a triangle respectively. The decision: does a given node define a division line or a triangle, depends on the neighbors of the nodes. Therefore, the computing time of this method can be given by $\mathcal{O}(|N|)$.

4.3 The Program *graphtomuq*

The task of the program `graphtomuq` is to create three-dimensional computer reconstructions from the muqarnas graph. It reads the structure information contained in the graph and defines a set of muqarnas elements (see Section 4.3.1). For visualizing the muqarnas, this set is converted into VRML-files containing the geometrical data of the muqarnas (see Section 4.3.2).

4.3.1 Calculation of the Muqarnas Structure

The following command line options are available to call the program `graphtomuq`:

- P**<string> This option is followed by the name of the file representing the complete muqarnas plan. If this option is not given, a subplan derived from the input muqarnas graph is used instead.
- G**<string> This option is necessary. It is followed by the name of the input file representing the muqarnas graph.
- p**<number> The option **-p** can be followed by the integers 1, 2 or 4, representing a full, half or quarter muqarnas plan, respectively. By default the input plan is considered being a quarter plan.
- v**<number> This option can be followed by 1 or 2, representing whether a full or half vault should be reconstructed.
- o**<string> This option expects the name of the directory in which the output files containing the geometrical data are written. By default, the output directory gets the name WRL.
- t**<string> This option is followed by the name of the texture file. By default the file `texture.jpg` is used.

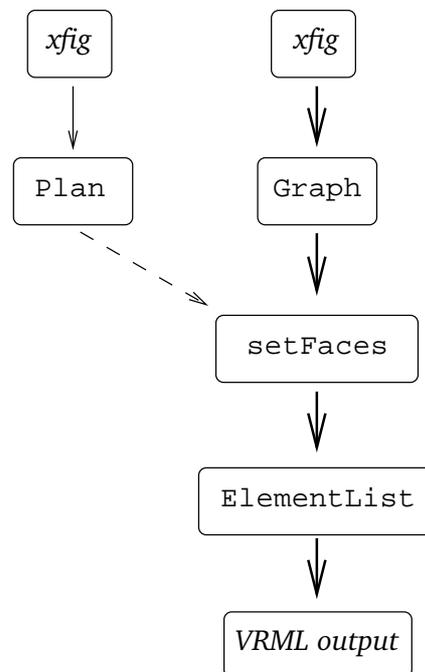


Figure 4.6. Program flow of the program `graphtomuq`.

- r With this option, we work with a data-structure where nodes in the plan are stored as floating-point numbers. By default we work with integers to represent the nodes in the graphs.

The program *graphtomuq* expects in the input the complete muqarnas plan, as well as the muqarnas graph. If the muqarnas plan consists only of basic elements we may use the subplan, derived from the muqarnas graph, instead of the complete plan. This subplan contains the same edges as the muqarnas graph, but without directions.

In Figure 4.6 the program flow of the program *graphtomuq* is visualized. The input consists of *xfig*-files which we convert into our own data-structure, like we also did in the program *plantograph*. We use the numbers behind the command line options -**p** and -**v** to mirror the input plan and graph so that they correspond to the complete projection of the muqarnas vault, which we want to reconstruct. For this mirroring, we first compare the coordinates of the nodes, to determine the maximal x and y coordinate of all nodes appearing in the plan. We iterate over the edges and calculate for each edge the mirror image of its end-nodes after mirroring in the x -mirrorline or y -mirrorline, respectively. The edge connecting the mirror images of the end-nodes is then inserted in the plan.

After mirroring, the graph and plan are stored in the object `setFaces` in which the conversion to the three-dimensional muqarnas vault takes place. This conversion is done by iterating over the nodes in the muqarnas graph. An extensive description is already given in Section 3.3. We recall that first the faces are recognized from the graph (see Section 3.3.2), and then these faces are converted into a list of elements, containing information about the type and the position (see Section 3.3.3).

When the muqarnas only consists of basic elements, it is possible to work with a simplified plan (see Section 3.3.4). In that case only the muqarnas graph is necessarily in the input. During the conversion to the three-dimensional muqarnas vault, we need to decide which rhombi and squares in the plane projection may need to be interpreted as two elements.

A rhombus will be interpreted as an almond and a biped if otherwise we need to insert a cell with diameter the longer diagonal of the rhombus. We take this into account during the conversion of the faces to elements. In step 4 of the algorithm: translate cell-faces to cells, we calculate the diameter d of the element. If this diameter is too large, namely larger than that of an almond, whereas the angle equals to $\pi/4$, we set this diameter equal to that of an almond, $d = 1/\cos(\pi/8)$. We insert the nodes belonging to the backsides given by this cell-face as nodes of an int-face. From this int-face a biped is created during step 5: translate int-faces to intermediate elements. We recognize this biped in the same way. We find an intermediate element with angle $\pi/4$, but diagonal d too larger for a biped. We set this diagonal equal to the length of the diagonal of a biped: $d = 1/(\sqrt{2}\cos(\pi/8))$.

We recall that we interpret a square as the plane projection of a biped and a jug, only if squares do not fit otherwise (see Figure 3.19). We recognize this situation by working during setting the faces with a plan `UsedBacksides`. In this plan we store the arrows used as backsides of the elements instead of curved sides. We then recognize the structure as given in Figure 3.19 as putting two squares would result in using arrow c two times as backsides of an element and not as a curved side.

4.3.2 Visualization of the Muqarnas in VRML

To visualize the geometry of the muqarnas we have chosen to use the VRML 2.0-format. This data-format gives a possibility to define three dimensional geometrical data. Different browsers supporting VRML 2.0 are available to visualize the data. We present here a short overview of the attributes of the VRML 2.0-format which we use to visualize the muqarnas, a detailed explanation of the VRML 2.0-format can be found in [Ames et al., 1997].

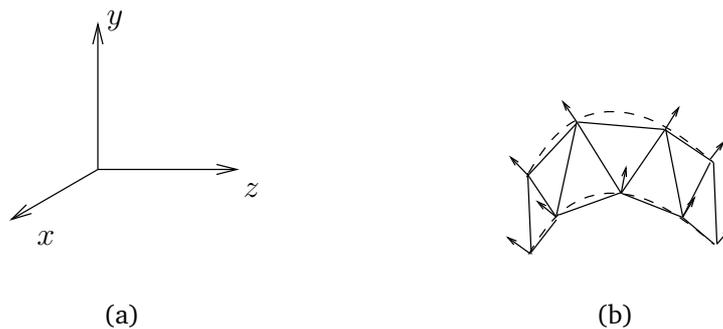


Figure 4.7. On the left: the xyz -coordinate system as defined in VRML. On the right: smooth surface (dotted lines) approximated by triangles, the arrows represent the normals vectors at the vertices on the surface.

In Figure 4.7(a) we see an illustration of the xyz -coordinate system in VRML. The x -axis points to the front, the y -axis points upwards and the z -axis points to the right. Surfaces in VRML are approximated by triangles (see Figure 4.7(b)). Besides of a list of coordinates (x, y, z) of vertices on the surface, a list of triangles is given. Each triangle is given by three numbers which refer to the selected vertices on the surface. In this way a triangulation of the surface is defined.

To give a smooth appearance to a surface constructed by joining triangles, we may also give a list of normal vectors explicitly (see e.g. [Foley et al., 1994, p. 541]). These normals are then used by the VRML-browser to calculate the amount of light to reflect in the different directions. This geometric information, coordinates, triangulation and normals, is stored in a so called `Geometric Node`.

To translate, rotate or rescale surfaces a `Transform Node` is available in VRML 2.0. It is used to change previous defined surfaces, or objects containing more surfaces, and place them on the required position in our model. These surfaces can be defined in the same file or we call to another file containing the geometrical data of the object we want to transform. For visualizing the muqarnas, we create separate VRML 2.0-files containing only the geometrical information of the muqarnas elements. By using `Transform Nodes` these elements are then placed in their right position.

Our output consists of three kind of VRML 2.0-files, files representing elements, files for the tiers and a main file binding the tiers to a muqarnas. For each muqarnas element, we create two files. In one of the files the roof of the element is stored and in the other the facets are stored (see Definition 2.1.1). This is done to have the freedom to rescale these parts of the elements separately (see Section 1.1.2). The files have names like *type_d_a_right_left_part.wrl*, where *type* can be `cell` or `int` to denote whether the file contains a cell or an intermediate element, *d* is a floating-point number which represents the diameter of the element, and *a* is a floating-point number to represent the angle of the element. The variables *right* and *left* contain the sizes of the right and left curved sides. The variable *part* can be equal to `u` or `l` for denoting if the file contains the upper or the lower part of the element. For example, the roof a cell with a square as plane projection is then given in the file `cell_1.41_1.57_1.0_1.0_u.wrl` and its facets are defined in `cell_1.41_1.57_1.0_1.0_l.wrl`.

The main file `main.wrl` consists of `Transform Nodes` putting the different tiers on each other. Each tier *i* is given by two files: `tieri_u.wrl` and `tieri_l.wrl`, representing the roof and the facet of the tier. In the tier files the elements are included by `Transform Nodes` containing the location and orientation of the elements.

4.3.3 Visualization of the Elements

We use for the visualization of our virtual reconstructions the curved muqarnas elements as described by al-Kāshī (see [Dold-Samplonius, 1992, p. 232]). This means that our reconstructions give a representation of the structure of the muqarnas, but do not need to resemble the original muqarnas. To create a virtual muqarnas resembling the real muqarnas, we may need to replace the curved elements by other kind of elements.

Our model for the curved muqarnas elements fit well to the elements found at the Takht-i-Sulaymān (see [Harb, 1978]). Because we focus on the structure and not on the static part of the muqarnas, in our model only the surfaces which are visible are drawn (see Figure 4.8). This means that we restrict ourselves by modeling the curved sides, the curved front surfaces and the bottoms of the elements, which are visible if we watch the muqarnas from beneath. The backside of an intermediate element only consists of the edge where its curved sides join and are therefore already reconstructed by drawing the curved sides. The backside of a cell is not visible in the muqarnas and hence not included in our model. The same holds for the top surfaces, as we do not look at the muqarnas from above they are not visible and therefore not drawn.

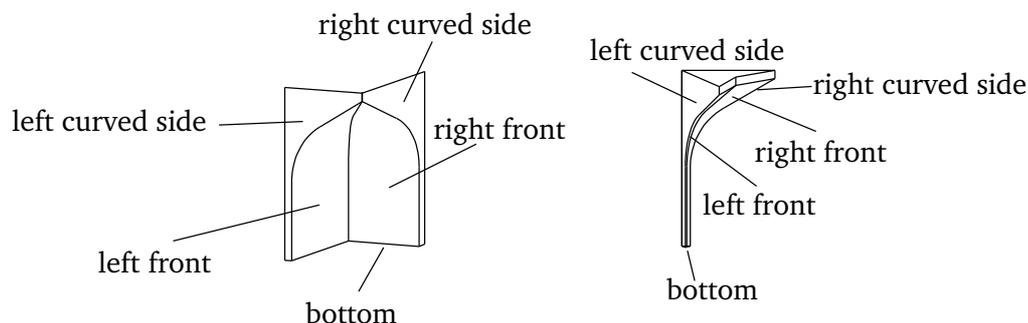


Figure 4.8. Element model as used for our virtual reconstructions.

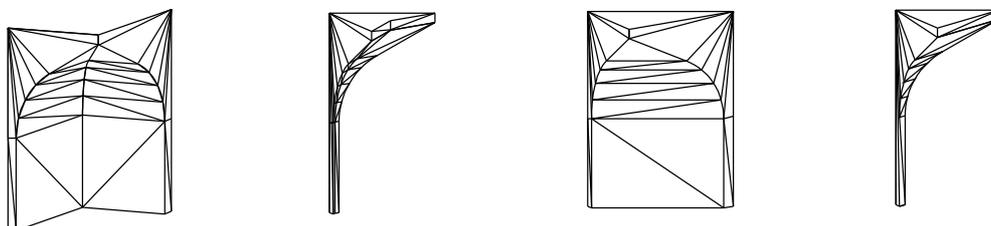


Figure 4.9. Elements and half-elements approximated by triangles.

In Figure 4.9 we see how we divide the surfaces of a cell and intermediate element in different triangles. These pictures show which coordinates we need to calculate for modeling the geometry of the elements. The curved side which represents the curved side of the elements is drawn in Figure 4.10. The curved side, as it is described by al-Kāshī, is given by straight lines and a circle arc. On the circle arc we calculate n vertices, where n depends on the smoothness and velocity of the reconstruction which we require for the drawing. By increasing n the reconstruction will be more realistic, but slower. This means that it takes longer until the reconstruction appears on the screen and it influences the speed with which we can move the muqarnas interactively.

The curve is described by al-Kāshī as according with the “method of the masons”, indicating that it is taken from practice. A detailed explanation of his calculation of the curve can be found in [Dold-Samplonius, 1996, p. 66]. An animation to illustrate his calculations can be found in [Dold-Samplonius et al., 2002].

We use the symbols as given in Figure 4.10 to describe the curve. Our origin is chosen so that vertex $G = (0, 0)$. Setting our unit equal to the module, we draw a straight line BG of length 2. The perpendicular line BA , which has size the module has then length 1. The oblique line AE intersects the opposite vertical BG with angle $\angle BAE$ equal to 30° . This line AE is divided in five equal parts, and on three fifth of this line we mark the vertex Z . As $\angle BAE = 30^\circ$, we find that E has coordinates $(0, 2 - 1/\sqrt{3})$ and it is then easy to verify that $Z = (\frac{2}{5}, 2 - \frac{1}{5}\sqrt{3})$. The line EZ is rotated down over E until it joins the vertical line. We find the end-point $H = (0, \frac{3}{5}\sqrt{3})$. The arc ZH is described by

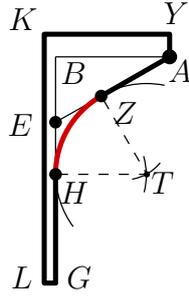


Figure 4.10. Construction of the curve according to al-Kāshī.

a circle with radius equal to the length ZH . To describe this arc we look for the center of the corresponding circle, which has distance ZH to H and to Z . This center can be found by intersecting the circle with center Z and radius ZH by the circle with center H and radius ZH . We find the center $T = (\frac{4}{5}, 2 - \frac{3}{5}\sqrt{3})$ and describe the arc ZH by

$$\varphi \mapsto \left(\frac{4}{5} + \frac{4}{5} \cos \varphi, \left(2 - \frac{3}{5}\sqrt{3}\right) + \frac{4}{5} \sin \varphi \right)$$

with $\frac{2}{3}\pi \leq \varphi \leq \pi$. The lengths of LG and AY vary, in our model we choose $LG = 0.1$ and $AY = 0.2$.

The coordinates of the curved side are embedded in the three-dimensional space by adding a z -coordinate equal to 0. We translate by $x = 0.1$ as we want to have vertex L in the origin instead of G . In the muqarnas reconstruction the module will be the length of KY and not of BA as an edge in the muqarnas plan corresponds to this length. Therefore we rescale by dividing the x -coordinates by 1.1. For the same reason we divide the y -coordinates by 2.1, that is the length of KL . As a consequence of this rescaling, the proportions of the curve change a little. The curve does not exactly coincides with the curve of al-Kāshī, but still serves as a good approximation. We list all coordinates of this curved side into a matrix curve.

For a cell given by $a = (a_x, 0)$, $b = (b_x, b_y) \in \mathbb{R}^2$ with diameter $d \in \mathbb{R}$ (see Definition 2.2.3 and Figure 2.15), we calculate the coordinates of the right curved side by translating over $(-1, 0, 0)$, so that the projection of the central node lies in the origin. To control the length of the curved side we rescale by multiplying the x -coordinates with $\|a\|$. The coordinates on the left curved side are given by translating the matrix curve over $(-1, 0, 0)$ and we rescale in the x -direction by multiplying the x -coordinate by $\|b\|$. Additionally, we multiply the matrix by the rotation matrix $R_{\alpha(0,a,b)}$ with

$$R_\varphi = \begin{pmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{pmatrix} \quad (4.4)$$

gives the coordinates of the vertices on the left curved side. The matrix R_φ defines a counter-clockwise rotation by an angle φ over the y -axis. To calculate the coordinates on the diagonal again we translate the coordinates as stored in the matrix `curve` over $(-1, 0, 0)$, then we multiply the x -coordinates of the matrix by the diameter d and rotate over $R_{\alpha(0,a,b)/2}$.

For an intermediate element given by $a = (a_x, 0), b = (b_x, b_y) \in \mathbb{R}^2$ and with diameter $d \in \mathbb{R}$, we calculate the coordinates similar. In this case the projection of K corresponds to the central node and a translation is not needed. The vertices on the right curved side are then calculated by multiplying the x -coordinates of the matrix `curve` by $\|a\|$. The vertices on the left curved side are calculated by multiplying the x -coordinates of the matrix `curve` by $\|b\|$ and rotating over the y -axis by $\alpha(0, a, b)$. The coordinates of the vertices on the diagonal are calculated by multiplying the x -coordinates by the diameter d and by multiplying the resulting matrix by $R_{\alpha(0,a,b)/2}$ for rotating purpose.

Calculation of the Normals

To give the viewer of our computer reconstructions the impression that the muqarnas elements are built with smooth surfaces instead of joined triangles, we explicitly write normals in the `Geometric Nodes` of the VRML 2.0-files. For each triangle we determine the normal vectors for its vertices, where not the normals of the triangle is given, but the normals of the surface we are approximating.

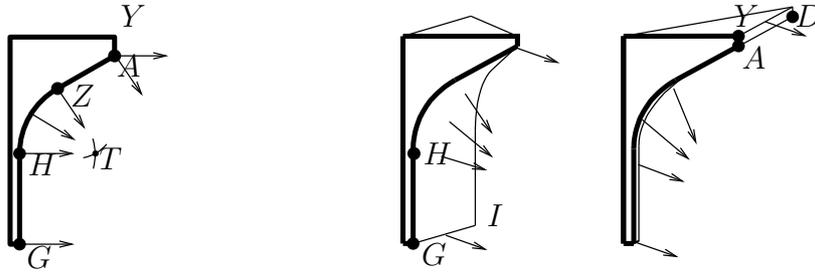


Figure 4.11. Calculation of the normals on the front surfaces of the elements.

In the calculation of the normals of the elements, we will use the normals on the curve as drawn in the most left picture of Figure 4.11. We consider the curved side as two-dimensional surface in the xy -plane and calculate its normals as vectors in this plane. For calculating normals for the front sides of the muqarnas elements we rotate normals of the curve adequately over the y -axis.

The normal of the vertices on the line YA are given by $(1, 0, 0)$ and the normals of the vertices on the line AZ are given by $(\frac{1}{2}, -\frac{1}{2}\sqrt{3}, 0)$. For the vertices v on the circle arc ZH , the normals n_v are given by

$$n_v = -\frac{v - T}{\|v - T\|},$$

as they point to the circle center T . The normals on the line GH are then given by $(1, 0, 0)$. We list our normals in a matrix `normals` and will use them for the front sides of the muqarnas elements.

We first consider the situation of a cell in standard position given by $a = (a_x, 0)$, $b = (b_x, b_y)$ and diameter d . Let $\alpha^* = \alpha(0, a, b)$ be the angle of the cell. We denote o for the vertex opposite of the central node. Its coordinates can be given by

$$o = -d(\sin(\alpha^*/2), -\cos(\alpha^*/2)).$$

We use again the notation R_φ to denote the rotation over the y -axis (see Equation (4.4)).

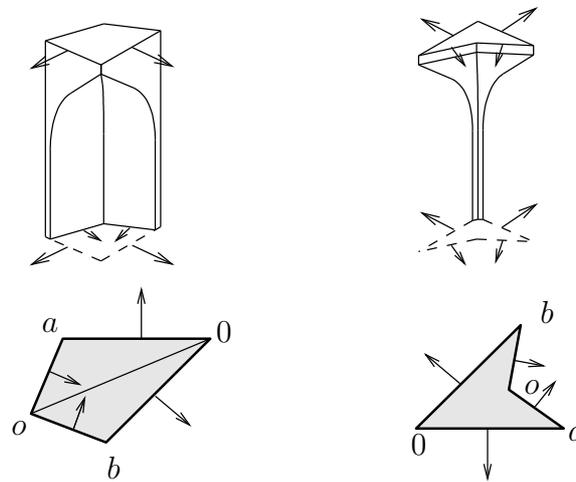


Figure 4.12. Calculation of normals on the surfaces in our element-model.

In the left picture of Figure 4.12 a cell is drawn with its projection given in standard position. The projection of the right curved side is given by $\text{edge}(a, 0)$. The normals of the vertices on this surface are then given by $(0, 0, -1)$. The projection of the left curved side is given by $\text{edge}(0, b)$. To decide its normals we rotate $(-1, 0, 0)$ over the y -axis with angle α^* so that it joins to $\text{edge}(0, b)$. Then we rotate over $\pi/2$ so that the normal is perpendicular to $\text{edge}(0, b)$. The normals of the vertices on the left curved side are thus given by

$$R_{\alpha^* + \pi/2} \cdot (-1, 0, 0).$$

The normals of the front sides are given by rotation of the matrix `normals`. The normals in this matrix are given for three-dimensional surfaces created by expanding the two-dimensional curve in three dimensions as the left two pictures of Figure 4.11. The normals as given in the matrix `normals` correspond to the normals of three dimensional surfaces perpendicular to $\text{edge}(a, 0)$, as drawn in the middle picture of Figure 4.11, with direction GI . For the front side, the projection of GI needs to join $\text{edge}(o, a)$. Therefore,

we rotate the matrix `normals` over $\pi/2$ so that it joins to `edge(a, 0)` and then rotate over $\alpha(a, 0, o)$. We find the normals of the vertices on the right front surfaces by

$$R_{\alpha(a,0,o)+\pi/2} \cdot \text{normals}.$$

Since `edge(a + b, b)` is parallel to `edge(a, 0)`, we find with similar arguments that the normals of the vertices on the left front surface can be given by

$$R_{\alpha(b,a+b,o)+\pi/2} \cdot \text{normals}.$$

For an intermediate element in standard position, given given by $a = (a_x, 0), b = (b_x, b_y) \in \mathbb{R}^2$ and with diameter $d \in \mathbb{R}$, we proceed in a similar way. Let $\alpha^* = \alpha(0, a, b)$ be the angle of the element, we write again o for the node opposite the central node, its coordinates are given by

$$o = d (\sin(\alpha^*/2), \cos(\alpha^*/2)).$$

With the same arguments we state that the normals of the vertices on the right curve are given by $(0, 0, 1)$, and the normals of the vertices on the left curved side are given by

$$R_{\alpha^*+\pi/2} \cdot (1, 0, 0).$$

For the front sides we can verify that the normals of the right front side are given by

$$R_{\alpha(b,a+b,o)+\pi/2} \cdot \text{normals},$$

and the normals of the left front side are given by

$$R_{\alpha(a,0,o)+\pi/2} \cdot \text{normals}.$$

4.3.4 Complexity Analysis

The structure information of the muqarnas is already contained in the muqarnas graph G . The task of the program `graphtomuq` is to translate this structure into a three-dimensional reconstruction. We find the faces by looking for each node at his neighbors. We conclude that the computing time to set the faces is linear in the number of nodes $|M|$ of the graph G .

To set the element type, we use operations with computing time not depending on the size of the graph. For the position only the calculation of the height depends on the size of the plan. For this we use Algorithm A-3.3. We have seen in Section 4.2.2 that its computing time depends on $|M|^3$, this cubic computing time is then crucial. The program `graphtomuq` needs at most $\mathcal{O}(|M|^3)$ operations.

Chapter 5

Results of Algorithmic Muqarnas Reconstructions

In this chapter we present the results of some examples of muqarnas plans on which we applied our algorithms to create computer reconstructions from the corresponding muqarnas vaults. We first consider examples of Seljuk muqarnas (see Section 5.1). In these examples, we use the data-structure for the plans and graphs, with the nodes stored as floating-point numbers (see Section 4.1.2). In the second part we consider examples of Il-Khanid muqarnas (see Section 5.2). Here we represent the nodes of the plans and graphs with integers (see Section 4.1.2).

For each muqarnas we first present a compact overview of properties of the muqarnas vault. This overview includes information about the location and building where the muqarnas vault can be found. The number of tiers the muqarnas consists of is also given. Considering this number during the reconstruction process may result in a restriction on the possible muqarnas reconstructions. Furthermore, the overview contains information about the time period in which the muqarnas vault probably was constructed. These years of construction are taken from the references presented in the overview. In the reference list, for most examples a reference to the website of Professor Shiro Takahashi (see [Takahashi, 2004]) is given. This refers to a big data-base of muqarnas plans available on the web. The muqarnas plans contained in this data-base are represented by a number denoted with $\#n$. We adopt this notation in our reference list.

After this overview, we discuss the results of applying the programs `plantograph` and `graphtomuq` on the muqarnas designs in order to reconstruct muqarnas vaults. In the examples, we always use a quarter muqarnas plan for the input. The plans are oriented as presented in Section 2.2.2. The origin is chosen so that the smallest x -value and y -value of the nodes appearing in the plan have coordinate 0. Most graphs are drawn with blue and red colors. If not stated otherwise, these colors are alternating for easier recognizing the tier information from the graphs.

5.1 Computer Reconstructions of Seljuk Muqarnas

The specialties of Seljuk muqarnas are the shell-shaped ornaments which we represent by textures. We use for our reconstructions of Seljuk muqarnas curved elements without facets, which means that we remove the lower parts of the elements. The combination of using elements without facets and applying textures gives our reconstructions a similar appearance as the still existing Seljuk muqarnas. We visited all sites of the examples of Seljuk muqarnas presented here.

5.1.1 Alay Han

<i>Location:</i>	On the road between Aksaray and Nevşehir (Turkey)
<i>Building:</i>	Alay Han
<i>Vault:</i>	Entrance portal to enclosure
<i>Height:</i>	7 tiers
<i>Years of construction:</i>	1180-1200
<i>Date of visit:</i>	April 9th, 2005
<i>References:</i>	[Erdmann, 1961, pp. 81–83], [Aslanapa, 1971, pp. 147–148], [Takahashi, 2004, #043]

Discussion

The road from Aksaray to Nevşehir crosses the area where once the courtyard of the Alay Han, the oldest known caravanseray in Anatolia, has been. From the courtyard less remained, but the entrance portal to the enclosure is in good state of preservation (see Figure 5.2(b)). It contains the oldest muqarnas vault we study in this work.

In Figure 5.1(a) the plan `pl_alayhan.fig` representing the plane projection of the muqarnas is shown. The muqarnas plan is mirror-symmetric in its diagonal. Furthermore, we find that the muqarnas is built mainly from basic elements. The elements are rotated over $\pi/8 + k\pi/4$, $k = 0, \dots, 7$ compared to standard position. Because of this property we can represent the nodes by integers, by remarking that the nodes in the plan are in the lattice \mathbb{L} rotated over $\pi/8$. Instead of rotating the lattice we use here for our reconstruction a data-structure based on floating-point numbers.

In Figure 5.1(b) the resulting subplan after running

```
removelines -r -Ppl_alayhan.fig
```

is shown. We find that indeed most lines not corresponding to curved sides are removed. The dotted lines in this figure represent edges we need to remove manually to create the subplan consisting only of the projection of the curved sides. Running

```
plantograph -r -Ppl_alayhan.fig -Qrmlines_pl_alayhan.fig
```

calculates the direction of several orbits of the muqarnas graph. There are four orbits of which the direction is undetermined (see Figure 5.1(c)). As the muqarnas plan is mirror-symmetric in its diagonal, we expect the two orbits marked red in Figure 5.1(d) having the same direction. The marked node then gives, by using Rule 4, the direction of these two orbits.

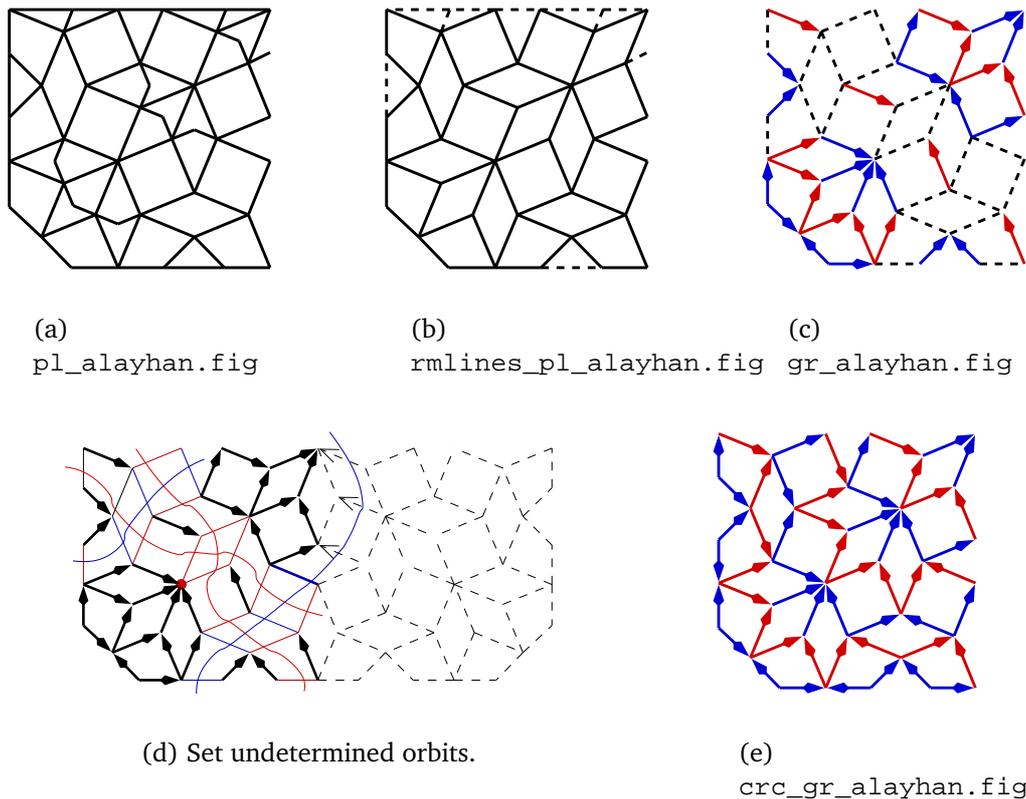


Figure 5.1. Reconstruction process for the muqarnas in the portal to the enclosure of the Alay Han.

For the blue marked orbits, we remark that the quarter plan only represents half of the plane projection of the muqarnas. If we copy the plan in the y -mirrorline we find that the direction of the most right blue orbit can be determined as the corresponding orbit in the full plan contains an edge incident to the center node. Rule 2 determines then its direction. To set the direction of the other blue marked orbit, we use the argument that the plan is mirror-symmetric in its diagonal, and therefore we expect the muqarnas graph being mirror-symmetric in its diagonal.

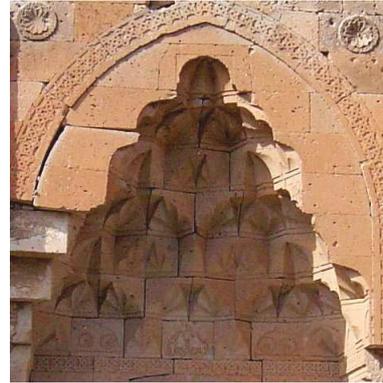
The muqarnas can then be reconstructed (see Figure 5.2(a)) by the command

```
graphtomuq -r -Ppl_alayhan.fig -Gcrc_gr_alayhan.fig.
```

We see in the reconstruction that there appear intermediate elements with almonds as plane projection. These elements carry cells with a biped as plane projection. These elements were not listed in our set of basic elements (see Section 2.1.1) as in Il-Khanid muqarnas only bipeds appear in the plan as plane projection of intermediate elements and almonds as the plane projection of cells.



(a) WRL/main.wrl



(b) Photo of the muqarnas.

Figure 5.2. Computer reconstruction and photo of the muqarnas vault in the portal to the enclosure of the Alay Han.

5.1.2 Kayseri, Çifte Medrese

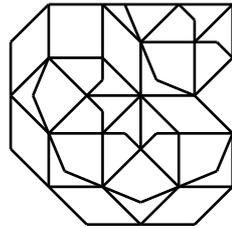
<i>Location:</i>	Kayseri (Turkey)
<i>Building:</i>	Çifte Medrese
<i>Vault:</i>	Entrance portal of hospital
<i>Height:</i>	7 tiers
<i>Year of construction:</i>	1205
<i>Date of visit:</i>	April 11th, 2005
<i>References:</i>	[Aslanapa, 1971, p. 129], [Takahashi, 2004, #040]

Discussion

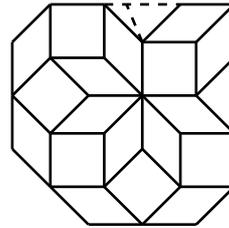
The Çifte Medrese in Kayseri consists of a medical school and a hospital which are joined by a corridor. It is the oldest hospital building in Anatolia. In Figure 5.3(a) a quarter muqarnas plan is given. It represents half of the plane projection of the muqarnas vault in the entrance portal of the hospital. The projection suggests that the muqarnas consists mainly of basic elements. Running

```
removelines -r -Ppl_ciftemed.fig
```

successfully removes edges which do not correspond to curved sides of muqarnas elements. The removed edges correspond to the projection of backsides of cells standing on front sides of intermediate elements. We need to remove additionally the edges which are given by dotted lines in Figure 5.3(b).



(a) pl_ciftemed.fig



(b) rmlines_pl_ciftemed.fig

Figure 5.3. Plan and subplan of the muqarnas in the entrance portal of the hospital of the Çifte Medrese in Kayseri.

By applying

```
plantograph -r -Qrmlines_pl_ciftemed.fig
```

the direction of all orbits can be determined (see Figure 5.4(a)). Calculating the corresponding muqarnas reconstruction results in the reconstruction as shown in Figure 5.4(b). Comparing this reconstruction with the muqarnas vault as given in Figure 5.4(g), we find that our virtual reconstruction does not fit to reality. This can, for example, be seen in the middle of the first tier: our virtual reconstruction has no element in the middle of the first tier, but the real muqarnas has. This is caused by the stone as shown in Figure 5.4(g) which is found in the middle of the muqarnas. It covers two tiers and does not correspond to a muqarnas element. If we remove by hand the lines from the plan, corresponding to this area (see Figure 5.4(c)), we are able to correct the corresponding muqarnas graph. The reconstruction calculated by

```
graphtomuq -r -Prmlines2_ciftemed.fig -Gcrc_gr2_ciftemed.fig
```

results in the reconstruction of the muqarnas with a gap on the place of the area not containing muqarnas elements. The effect of removing these edges is that the elements on tier 2 and 3 are pushed down so that place for the stone is created.

Like we have seen in the reconstruction of the Alay Han (see Section 5.1.1) also in this muqarnas cells with plane projection a biped standing on intermediate elements with plane projection an almond appear.

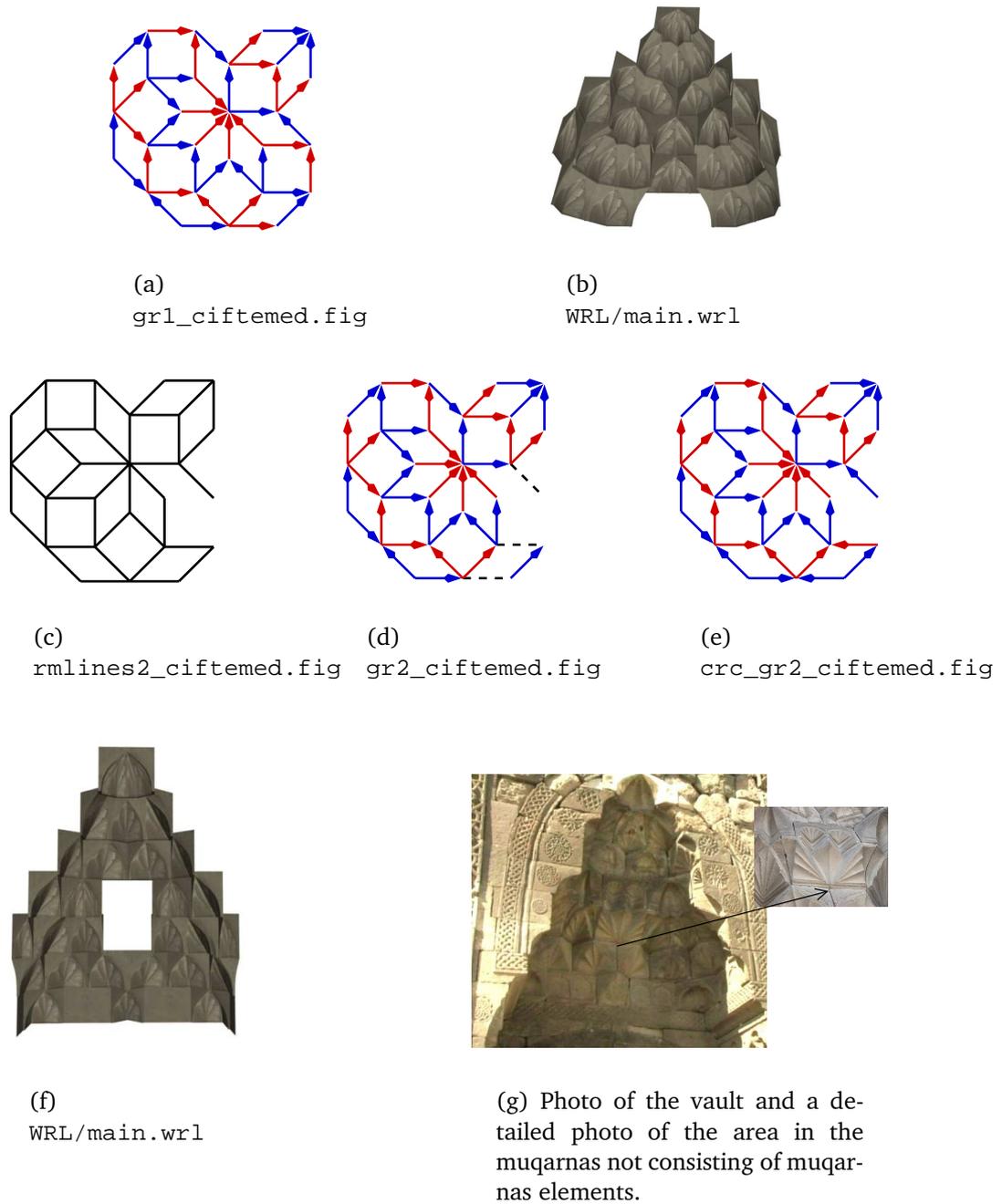


Figure 5.4. Reconstruction process and photos of the entrance portal of the hospital of the Çifte Medrese in Kayseri.

5.1.3 Sultan Han near Kayseri

<i>Location:</i>	On the road between Kayseri and Sivas (Turkey)
<i>Building:</i>	Sultan Han
<i>Vault:</i>	Entrance portal to enclosure
<i>Height:</i>	9 tiers
<i>Years of construction:</i>	1232-1236
<i>Date of Visit:</i>	April 11th, 2005
<i>References:</i>	[Erdmann, 1961, pp. 90–97], [Aslanapa, 1971, p. 150], [Takahashi, 2004, #051]

Discussion

The Sultan Han near Kayseri is in a good condition and richly ornamented with muqarnas. We study here the muqarnas in the entrance portal to the enclosure. In Figure 5.5(a) the plane projection of this muqarnas vault is given. We start the reconstruction process by removing edges edges of the plan by applying

```
removelines -r -Ppl_slthan_ksi.fig.
```

Additionally, the edges represented by dotted lines in Figure 5.5(b) need to be removed to create the subplan consisting of the projection of the curved sides. The muqarnas graph is calculated by running

```
plantograph -r -Qrmlines_pl_slthan_ksi.fig -Ppl_slthan_ksi.fig.
```

There are eight orbits for which the direction is not set. The direction of the undetermined orbits can be determined by using similar arguments as we did by reconstructing the Alay Han (see Section 5.1.1).

In Figure 5.5(d) the complete muqarnas graph is given. In this figure we colored one square gray. The direction of the arrows incident to this square show the motivation for excluding the edges of a polygon with cross edge from being opposite (see Definition 3.1.6). This part corresponds in the muqarnas to two half intermediate elements with plane projection half squares which join in their front, as can be seen by comparing this part to the right picture of Figure 5.6(b).

The computer reconstruction is given by

```
graphtomuq -r -Ppl_slthan_ksi.fig -Gcrc_gr_slthan_ksi.fig.
```

We see that there is a small part in the muqarnas not consisting of muqarnas elements. In our reconstruction a gap is left, in the real muqarnas this space is filled with a stone ornamented with a kind of rosette (see left of Figure 5.6(b)). Our computer reconstruction contains 10 tiers, but the real muqarnas vault contains only 9 tiers. The eight tier of the computer reconstruction only contains cells with projection rhombi. In the real

muqarnas the rhombi are the projection of intermediate elements of the seventh tier. It is an example of a local different muqarnas as described Section 3.5. If a tier only consists of cells standing on curved sides of elements in the previous tier, we can push down the complete tier.

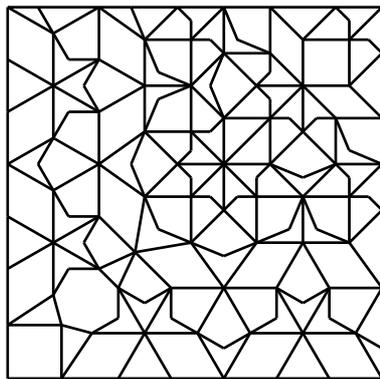
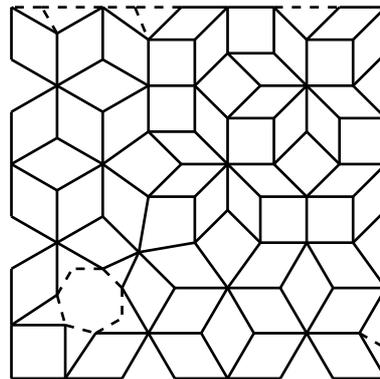
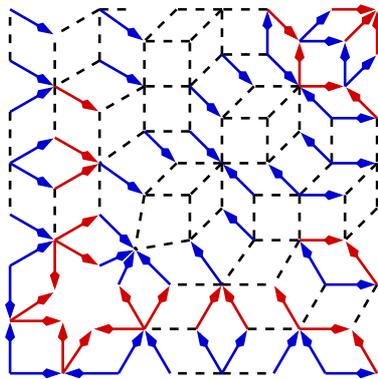
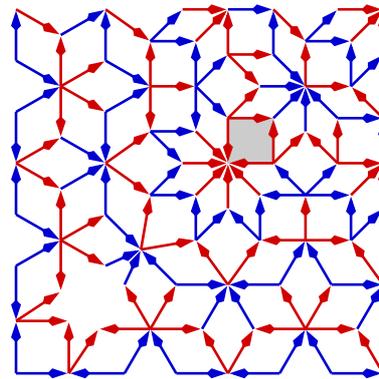
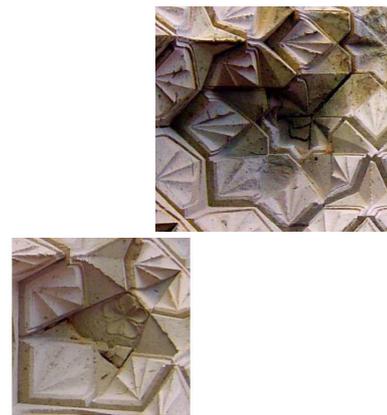
(a) `pl_slthan_ksi.fig`(b) `rmlines_pl_slthan_ksi.fig`(c) `gr_slthan_ksi.fig`(d) `crc_gr_slthan_ksi.fig`

Figure 5.5. Plan analysis for reconstructing the muqarnas vault in the entrance portal to the enclosure of the Sultan Han near Kayseri.



(a) Photo of front view.



(b) Photos of some details.



(c) Bottom view of the computer reconstruction.



(d) Photo of bottom view.

Figure 5.6. Pictures and reconstruction of the entrance portal to the enclosure of the Sultan Han near Kayseri.

5.1.4 Ankara, Arslanhane Camii

<i>Location:</i>	Ankara (Turkey)
<i>Building:</i>	Arslanhane Camii (Mosque)
<i>Vault:</i>	Entrance portal
<i>Height:</i>	9 tiers
<i>Year of construction:</i>	1290
<i>Date of visit:</i>	April 7th, 2005
<i>References:</i>	[Aslanapa, 1971, pp. 121–122], [Takahashi, 2004, #882]

Discussion

The last example we study of a muqarnas in Seljuk style, is the vault in the entrance portal of the Arslanhane Mosque in Ankara. In Figure 5.7(a) the plan of this muqarnas is shown. By removing edges not corresponding to curved sides of elements, we find that the subplan mainly consists of squares and rhombi (see Figure 5.7(b)). Comparing to the plans from muqarnas of the Il-Khanid period, we see that the plan is contained in the lattice \mathbb{L} rotated over $\pi/8$, like we also have seen in the reconstruction of the vault at the Alay Han (see Section 5.1.1). In the center elements appear of smaller size. In Figure 5.7(c) the output after running

```
plantograph -r -Qrmlines_pl_arslanhane.fig -Ppl_arslanhane.fig
```

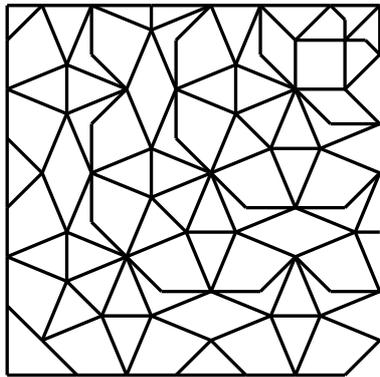
is given. This output is not satisfying, as too less directions of edges are set. If we look to the reason, we conclude that a lot of orbits in this plan only contain a few edges, due to the fact that most rhombi are divided in two triangles. In the definition of opposite edges (see Definition 3.1.6), we excluded edges in polygons with cross edges as being opposite. This was done for the case that the triangles should be interpreted as two half intermediate elements joining at their front, like we have seen in the reconstruction of the entrance portal of the enclosure of the Sultan Han near Kayseri (see Section 5.1.3). In the real muqarnas (see Figure 5.8(b)) such element combinations do not appear. The divided rhombi are the plane projections of a half cell standing on a half intermediate element, both elements with plane projection a half rhombus. It is a similar situation of a rhombus dividing in an almond and a biped (see Section 3.3.4). By running

```
plantograph -r -Qrmlines_pl_arslanhane.fig,
```

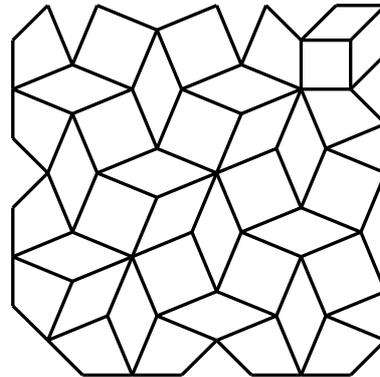
the complete plan is not given in the input and the information of the cross edges dividing the rhombi is not included. Therefore, the edges of the rhombi are interpreted as opposite edges. In this way we do not regard element combinations consisting of half intermediate elements joining at their front. In the output graph (see Figure 5.7(d)) much more directions are determined. The remaining directions are given manually (see Figure 5.8(a)) and the reconstruction is created by running

graphtomuq -r -Ppl_arslanhane.fig -Gcrc_gr_arslanhane.fig.

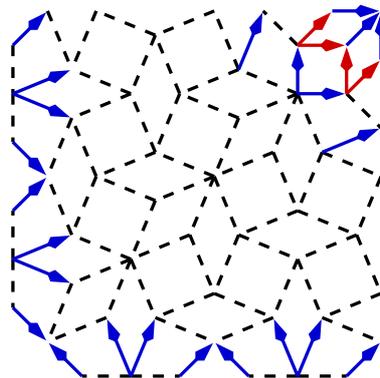
In Figure 5.8(c) the reconstruction is shown after manual completion of the vault the muqarnas fits in.



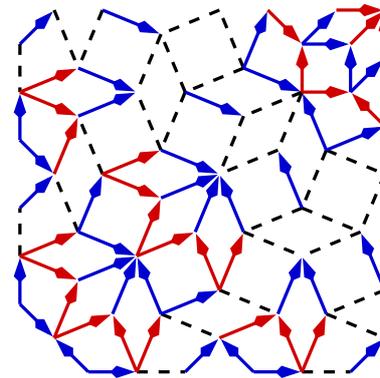
(a) pl_arslanhane.fig



(b)
rmlines_pl_arslanhane.fig

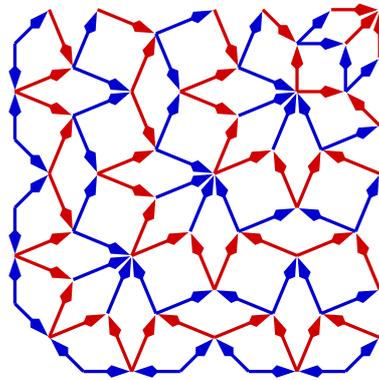


(c)
gr1_arslanhane.fig



(d)
gr2_arslanhane.fig

Figure 5.7. Plan analysis of the muqarnas vault in the entrance portal of the Arslanhane Mosque in Ankara.



(a)
crc_gr_arslanhane.fig



(b) Photo of the entrance portal.



(c) WRL/main.wrl

Figure 5.8. Reconstruction of the entrance portal of the Arslanhane Mosque in Ankara.

5.2 Computer Reconstructions of Il-Khanid Muqarnas

In the reconstructions of Il-Khanid muqarnas, we take into account that these muqarnas exist mainly of basic elements. We represent the plans and graphs with the data-structure based on integers. In the examples we set our unit equal to the module, the length of the side of a square. For the visualization we use a texture which is inspired by the appearance of a muqarnas vault in the dome of the Nur al-Din Mausoleum in Damascus, Syria. We have chosen this texture to emphasize the curves on the elements. In this way it is easier to recognize the different elements separately.

5.2.1 Natanz, Friday Mosque

<i>Location:</i>	Natanz (Iran)
<i>Building:</i>	Friday Mosque
<i>Vault:</i>	Basement of the north iwan
<i>Height:</i>	5 tiers
<i>Years of construction:</i>	1304–1309
<i>References:</i>	[Wilber, 1955, pp. 133–134], [Harb, 1978, pp. 54–55], [Takahashi, 2004, #120]

Discussion

The Friday Mosque in Natanz is in a pretty good state. In the north iwan two muqarnas placed above each other can be found. We study the lowest one and refer to it with “the muqarnas in the basement vault”. The plane projection of this muqarnas vault is given in the file `pl_natanz.fig` (see Figure 5.9(a)). By running the program `removelines` without the optional command line option `-h`, we remove the edges of length unequal to 1, not corresponding to element sides at the front of the vault. The resulting subplan `rmlines_pl_natanz.fig` is given as output (see Figure 5.9(b)).

The possible muqarnas graph corresponding to this subplan is calculated by applying

```
plantograph -Qrmlines_pl_natanz.fig,
```

the output is shown in Figure 5.9(c). Almost all directions of the edges are set by the program `plantograph`. There is only one direction undetermined (dotted edge in Figure 5.9(c)). By copying the plan in the y -mirrorline, this edge belongs to an orbit of which the direction is set. It is therefore obvious what the direction of this edge should be and we set it manual (see Figure 5.9(d)). By calling to `graphtomuq` a computer reconstruction is given, containing 7 tiers (see Figure 5.9(e)).

The muqarnas vault in Natanz contains 5 tiers and not 7, therefore we call

```
removelines -h5 -Ppl_natanz.fig.
```

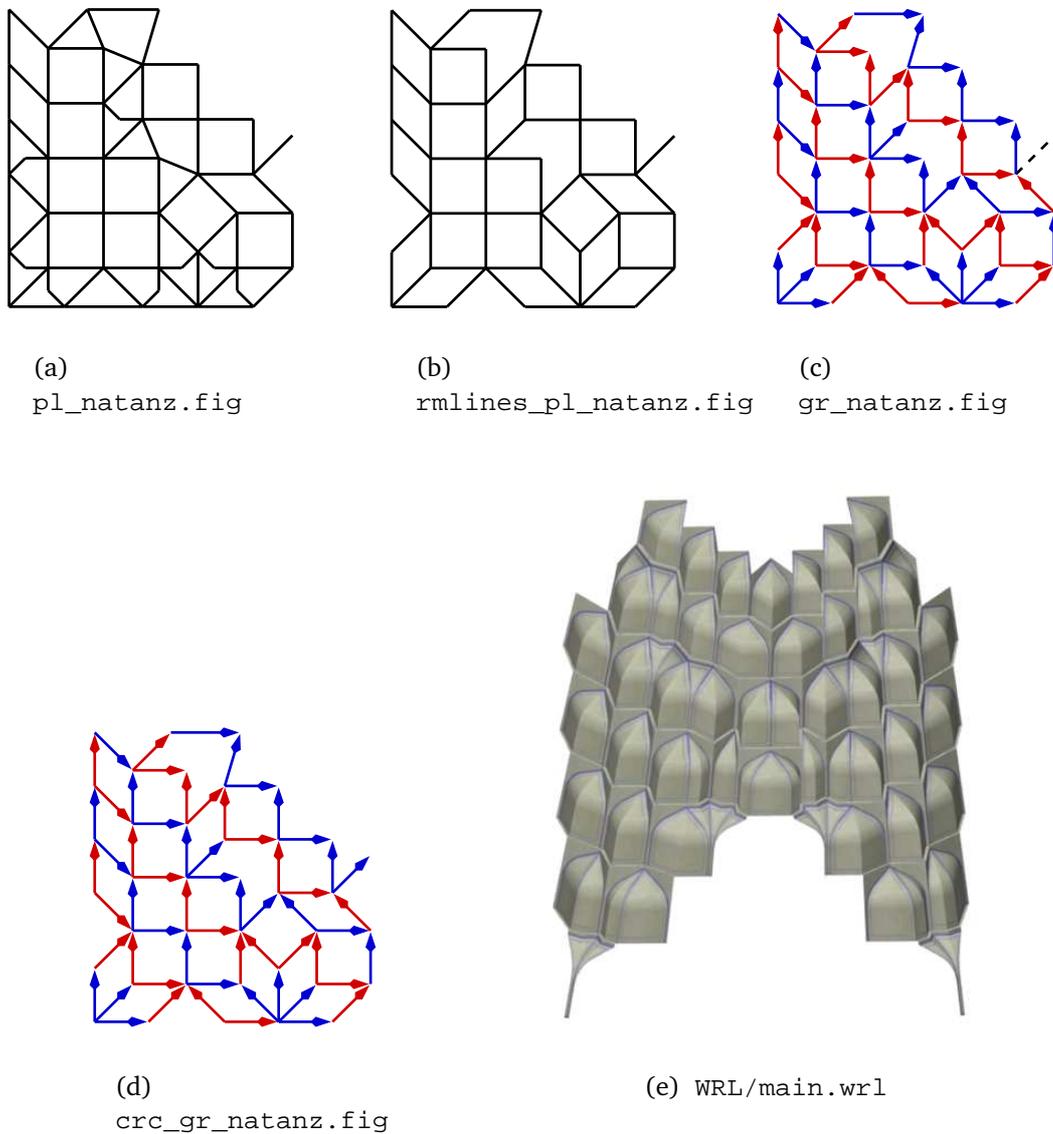


Figure 5.9. Plan analysis for reconstructing the muqarnas in the basement of the north iwan of the Friday Mosque in Natanz, without specifications concerning the height used for reconstruction.

This program states that reconstructing a muqarnas containing 5 tiers is possible by removing the node indexed by 312, which has coordinates $(1+1/\sqrt{2}, 1+1/\sqrt{2})$. We remove this node by hand.

We examine the effect of our rules (see Section 3.2) applied on this subplan by calling

`plantograph -Qrmlines2_pl_natanz.fig.`

In Figure 5.10(a) the result of applying Rule 1, the arrows at the bottom boundary point

inwards, in Figure 5.10(b) the result of applying Rule 2, the arrows point to the center and in Figure 5.10(c) the result of applying Rule 3, directions of arrows in an islands are known, is shown. Rule 4, there are no niches allowed, has no effect in this example. The computer reconstruction corresponding to the new graph (see Figure 5.10(e)) indeed agrees with the muqarnas in the basement of the north iwan of the Friday Mosque in Natanz.

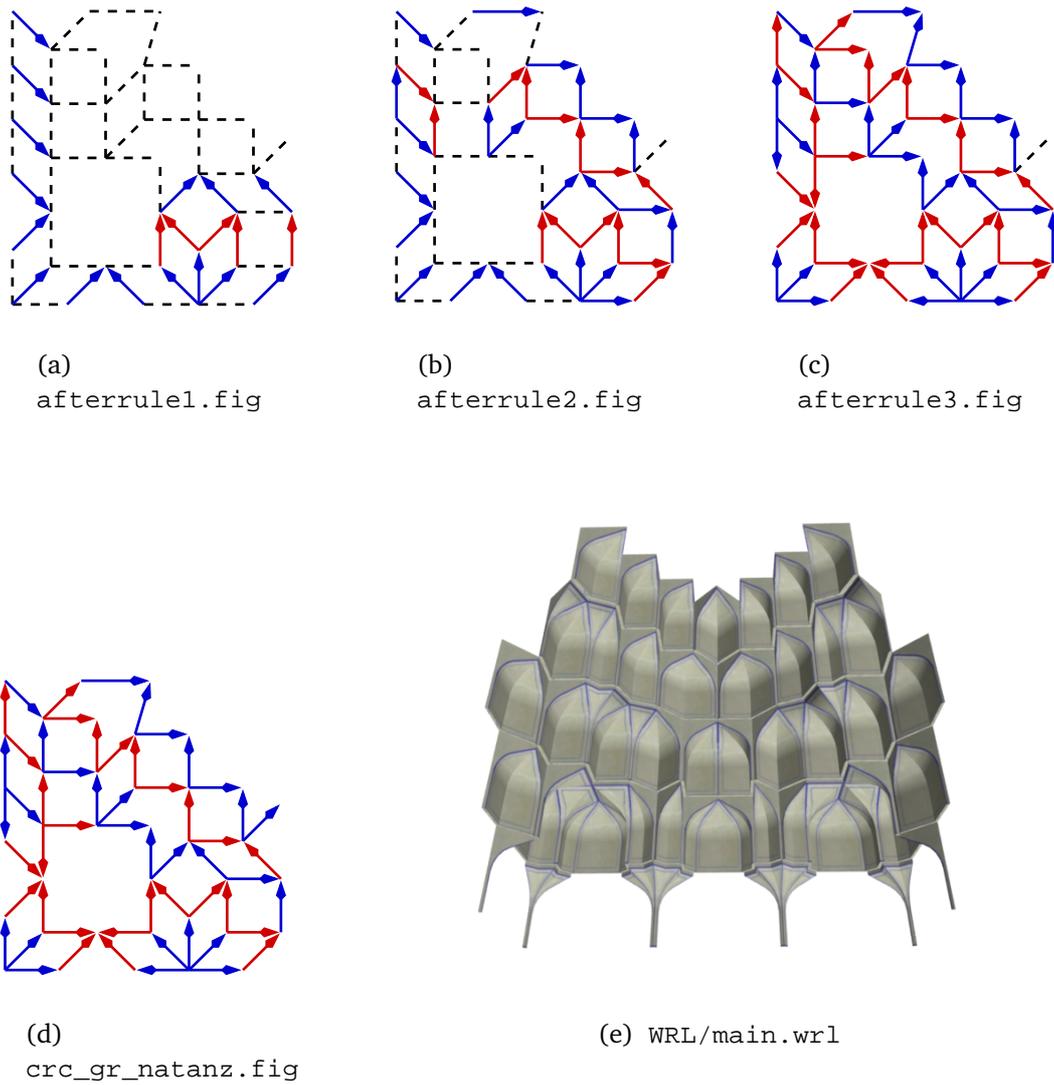


Figure 5.10. Plan analysis of the muqarnas in the basement of the north iwan of the Friday Mosque in Natanz, with node removal on the diagonal allowed to get desired height.

5.2.2 Biṣṭām, Shrine of Bāyazīd

<i>Location:</i>	Biṣṭām (Iran)
<i>Building:</i>	Shrine of Bāyazīd
<i>Vault:</i>	East portal
<i>Height:</i>	6 tiers
<i>Year of construction:</i>	1313
<i>References:</i>	[Wilber, 1955, pp. 127], [Harb, 1978, pp. 44–45], [Takahashi, 2004, #058]

Discussion

In the small village of Biṣṭām, which lies between Tehran and Mashad, two architectural elements dated in the Il-Khanid period can be found. These are the shrine of Bāyazīd and the Friday Mosque. We study here a muqarnas vault of the shrine of Bāyazīd.

Inspired by the design on the plate found at the Takht-i-Sulaymān, we use in this example a simplified plan (see Section 3.3.4). This means that lines which split a rhombus into an almond and small biped and lines which split a square into a jug and large biped are removed. As a consequence a rhombus in the plan may correspond to two elements: a cell with plane projection an almond standing on an intermediate element with plane projection a small biped. For the squares in the plan a similar situation is valid. A square may correspond to a cell with plane projection a jug standing on an intermediate element with plane projection a biped. The program `graphtomuq` should to be able to decide which rhombi need to be split. It is not able to recognize all squares which need to be split, but the combinations of squares where splitting is necessary are recognized.

As it is known that the original vault consists of 6 tiers we apply

```
removelines -h6 Ppl_bistam.fig.
```

This method removes the edges with length unequal to one and suggests to remove the node indexed by 612, which has coordinates $(2 + 1/\sqrt{2}, 2 + 1/\sqrt{2})$. In Figure 5.11(a) the removed edges are drawn with dotted lines. The directions of the edges is calculated by applying

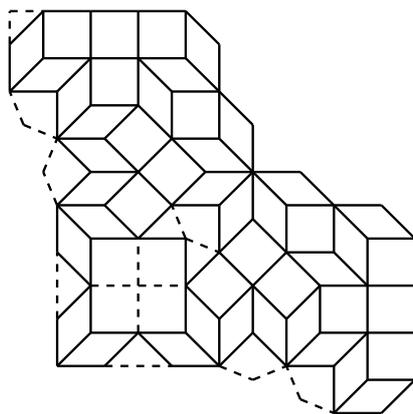
```
plantograph -Qrmlines_pl_bistam.fig
```

and calculating the computer reconstruction of the vault can be done by running

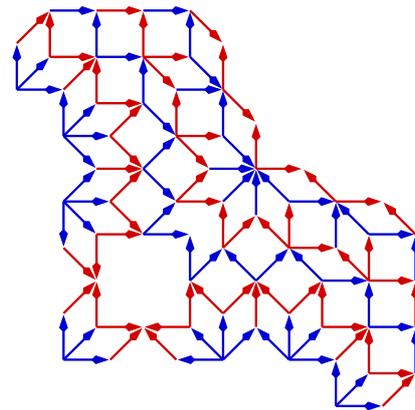
```
graphtomuq -Ggr_bistam.fig.
```

We see that the rhombi incident to the bottom boundary of the plan are indeed in the computer reconstruction of the muqarnas interpreted as almonds and bipeds covering the first two tiers. The squares are not split in the computer reconstruction. In the real muqarnas they are split, as we see in the plane projection of the existing vault as given in

Figure 5.12. Another difference between the original vault and our reconstruction are the half squares which appear in the second tier of our reconstruction. In the real muqarnas there are no elements on these places. The differences between the virtual muqarnas and the real one are only local. We need can exchange some cells with projection a square by a combination of a cell with projection a jug and an intermediate element with projection a large biped. The half elements at the second tier with projections half squares need also to be removed for creating a more realistic muqarnas reconstruction.



(a) pl_bistam.fig



(b) gr_bistam.fig



(c) WRL/main.wrl

Figure 5.11. Plan analysis for reconstructing the muqarnas in the east portal of the shrine of Bāyazīd in Bištām.

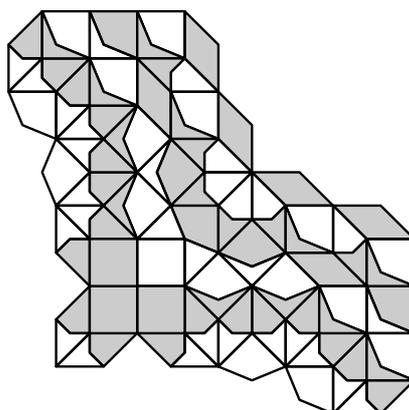


Figure 5.12. Plane projection with tiers colored alternately of the muqarnas in the east portal of the shrine of Bāyazīd in Bisṭām.

5.2.3 Farūmad, Friday Mosque

<i>Location:</i>	Farūmad (Iran)
<i>Building:</i>	Friday Mosque
<i>Vault:</i>	Sanctuary iwan
<i>Height:</i>	6 tiers
<i>Year of construction:</i>	1320
<i>References:</i>	[Wilber, 1955, pp. 156–157], [Takahashi, 2004, #059]

Discussion

In this example, we explain the effect on the muqarnas if we set the directions of undetermined edges in different ways. We observe a muqarnas from the Friday Mosque in Farūmad. This mosque is situated on the outskirts of the village. The earliest construction work at the mosque stems from Seljuk times, the damaged north and south iwan are from Il-Khanid time.

The plan of the muqarnas vault is given in Figure 5.13(a). We delete edges by applying

```
removelines -h6 -Ppl_farumad.fig.
```

It removes the edges with length unequal to 1 and proposes to remove the node on the diagonal with integer representation 364, which has coordinates $(1 + 3/\sqrt{2}, 1 + 3/\sqrt{2})$. The edges which we remove are represented in Figure 5.13(a) with dotted lines. The muqarnas graph is calculated by first applying

```
plantograph -Qpl_farumad.fig,
```

we find that most directions can be determined. The edges of the undetermined orbits can be set by using the fact that in the plan some squares are divided in jugs and bipeds. For these squares the direction of its edges is fixed by the orientation of the jug. This forces the direction of the undetermined edge and we get the graph as shown in Figure 5.14(a) with reconstruction calculated by applying

```
graphtomuq -Ppl_farumad.fig -Ggr1_farumad.fig
```

given in Figure 5.14(b).

We study what would happen if we set the direction of the undetermined orbits of Figure 5.13(b) in the reverse order. This graph is shown in Figure 5.14(c) and the corresponding reconstruction is created by applying

```
graphtomuq -Ppl_farumad.fig -Ggr_farumad.fig.
```

We see that this second reconstruction has a regular center, but the bottom boundary is not regular anymore. We find that the elements corresponding to the orbits of which we changed the direction are pushed upwards. This idea can be used to create several globally different muqarnas of different heights. Changing the direction of orbits makes it possible to push parts upwards or downwards.

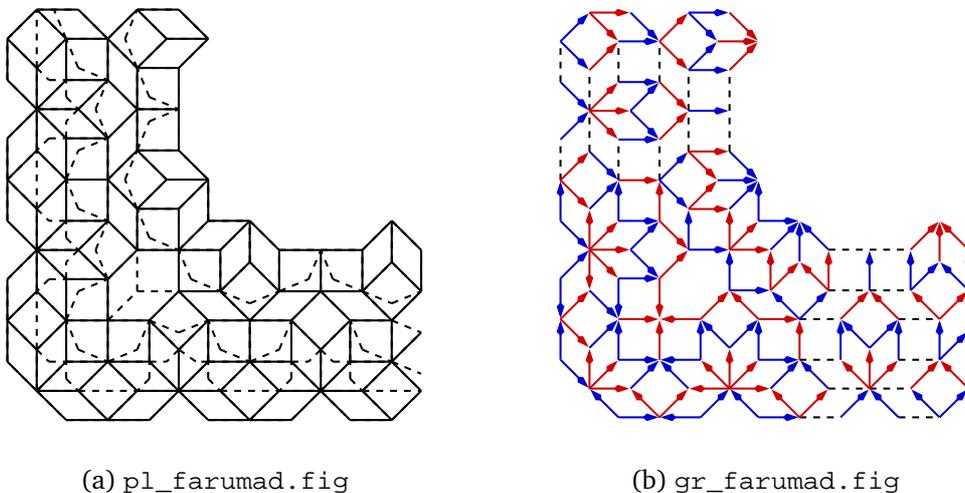
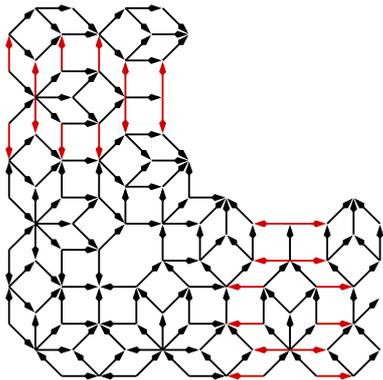


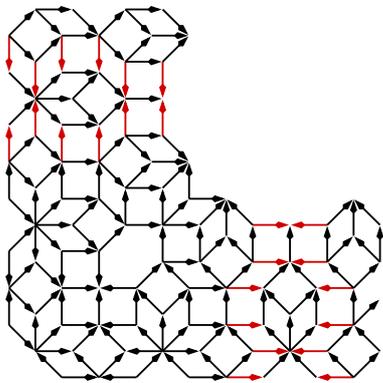
Figure 5.13. Plan analysis for reconstructing the muqarnas in the sanctuary iwan of the Friday Mosque in Farūmad.



(a) gr1_farumad.fig



(b) rec1_farumad.fig



(c) gr2_farumad.fig



(d) rec2_farumad.fig

Figure 5.14. Two different computer reconstructions corresponding to the simplified plan of the sanctuary iwan of the Friday Mosque in Farūmad.

5.2.4 Takht-i-Sulaymān, South Octagon

<i>Location:</i>	Takht-i-Sulaymān (Iran)
<i>Building:</i>	Il-Khanid palace
<i>Vault:</i>	South octagon
<i>Height:</i>	unknown
<i>Years of construction:</i>	1271–1274
<i>References:</i>	[Harb, 1978, pp. 43–46]

Discussion

In the Il-Khanid palace at the Takht-i-Sulaymān parts of a muqarnas were found in a room with octagonal ground plan. The palace contained two rooms with octagonal ground plan and the muqarnas remains were found in the most south one. Adopted from Harb, we will refer to this room as the *south octagon*.

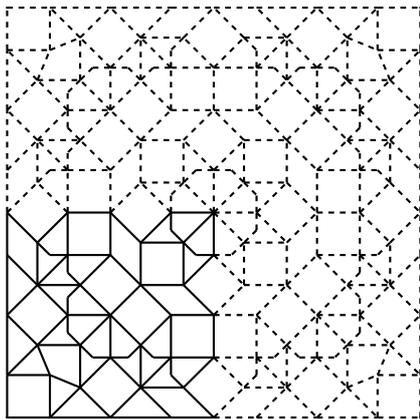
In Figure 5.15(b) a computer reconstruction of the complete muqarnas, as proposed by Harb is given, together with its plane projection in Figure 5.15(a). This reconstruction consists of 6 tiers. The corresponding muqarnas graph is a disconnected graph (see Figure 5.15(c)).

Because the graph is disconnected, we are not able to reconstruct the muqarnas by applying

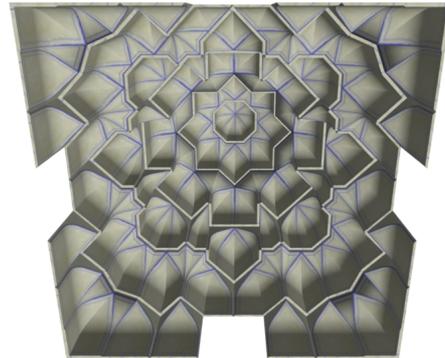
```
graphtomuq -Ppl_tis_southoctagon -Ggr_tis_southoctagon.
```

Our tier algorithm (see Section 3.1, Algorithm A-3.3) cannot be applied. We need to reconstruct the center and the outer part separately (see Figure 5.15(d)) and include the half rhombi in the fourth tier manually.

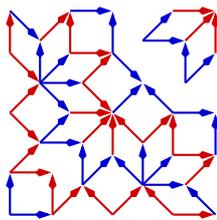
If we calculate a muqarnas reconstruction with the program `plantograph` from the plan, by calculating the muqarnas graph without removing edges (see Figure 5.16(b)), we get as corresponding muqarnas almost the same reconstruction. This reconstruction contains 7 tiers. The rhombus which Harb interpreted as intermediate elements appearing in the fourth tier, appears in this new reconstruction as cells in the fifth tier. The other elements are not changed. We see that this means only a local change and the global muqarnas is not influenced (see Section 3.5).



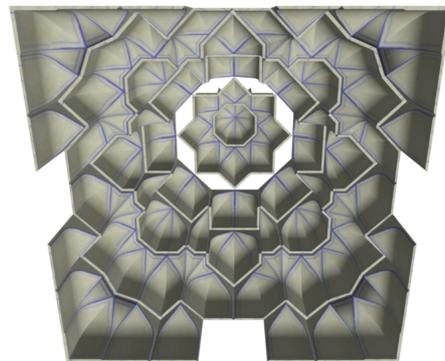
(a)
pl_tis_southoctagon.fig



(b) Reconstruction proposal of Harb.



(c) Muqarnas graph of Harb's reconstruction proposal.



(d) Reconstruction by using graphtomuq.

Figure 5.15. Reconstruction proposal of Harb for the muqarnas in the southoctagon of the Il-Khanid palace at Takht-i-Sulaymān.

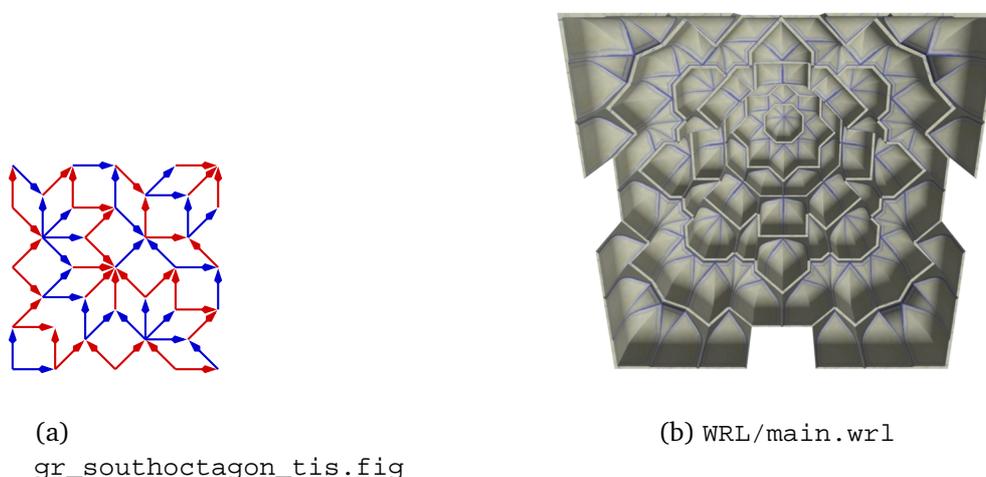


Figure 5.16. Reconstruction proposal of the muqarnas in the south octagon by using `plantograph` and `graphtomuq`.

5.2.5 Takht-i-Sulaymān, Plate

<i>Location:</i>	Takht-i-Sulaymān (Iran)
<i>Building:</i>	Il-Khanid palace
<i>Vault:</i>	Unknown
<i>Height:</i>	Unknown
<i>Years of construction:</i>	1271–1276
<i>References:</i>	[Harb, 1978, pp. 60–66], [Yaghan, 2000], [Dold-Samplonius and Harmsen, 2005]

Discussion

The most important example we consider in this work is the reconstruction of a muqarnas corresponding to the design on the plate found at the Takht-i-Sulaymān. We do not know how the muqarnas corresponding to this plate looked like, not even the proportions of the room for which the muqarnas was designed. It could be that it was only a sketch and the designed muqarnas was never built. We will see that our reconstruction methods succeed for understanding the design and results in different suggestions for reconstruction.

In the last decades different interpretations of the plate were published. The first interpretation was done by Harb (see [Harb, 1978]). Yaghan has published four different suggestions for reconstruction (see [Yaghan, 2000]) and we published another proposal

(see [Dold-Samplonius and Harmsen, 2005]). These reconstructions were mainly found by trial and error. Here we present our more methodical way for understanding the plate and creating computer reconstructions.

We first want to emphasize that it is unknown whether the design was correct. On the plate different lines emerge in the center of the design. It could be that the designer was uncertain about the design for the upper most tiers and tried different possibilities. To take in account these problems during the reconstruction process, we use the alternative Rule 2 by applying the command line option `-c` for calculating the muqarnas graphs.

In Figure 5.17(a) the muqarnas plan based on the design as recognized by Harb is given. The only difference between the plan and the design is that we changed the inner lines of the hexagons on the diagonal (see [Dold-Samplonius and Harmsen, 2005]) so that the design only consists of projections of basic elements. Comparing the diagonal of the plan in Figure 1.3 to the diagonal of the plan in Figure 5.17(a) shows the distinction. It is only a local difference and does not influence the global reconstruction. We could replace the corresponding elements in the reconstruction by elements according to the design on the plate without changing other elements in the muqarnas reconstruction.

By applying

```
plantograph -c-Qpl_tis_plate.fig
```

almost all directions can be determined (see Figure 5.17(b)). The resulting graph looks quite similar to the muqarnas graph corresponding to the proposal for reconstruction given by Harb. We first set the direction of the remaining edges of this graph according to Harb's reconstruction (see Figure 5.18(b)) and run

```
graphtomuq -v1 -G gr_tis_plate_harb.fig -Ppl_tis_plate.fig
```

to create a computer reconstruction of Harb's interpretation of the design. The plane projection of this reconstruction is given in Figure 5.18(a) where the tiers are colored alternately. Figure 5.18(c) shows the three-dimensional reconstruction which exists of 18 tiers.

For creating a reconstruction with regular center, the directions of the remaining edges are set by running `plantograph` without the command line option `-c`. They are then directed as shown in Figure 5.19(b). It results in another computer reconstruction, containing 17 tiers (see Figure 5.19(c)).

We prefer a muqarnas with a regular bottom boundary. Without removing edges this is not possible: independent of how we set the direction of the undetermined edges in Figure 5.17(b), the corresponding reconstruction has a non-regular bottom boundary. We try to find other reconstruction possibilities by applying the program `removelines`. A reconstruction with regular boundary contains supposedly less tiers, as a path from the corner of the muqarnas plan to the center needs to have the same length as a path starting at another location of the bottom boundary to the center (see Section 3.4). We therefore watch what happens if we try to remove edges in order to create a muqarnas reconstruction with maximal 16 tiers. We run

```
removelines -h16 -Ppl_plate_tis.fig.
```

The textual part of the output contains that we need to remove one or more of the nodes indexed by 144 , 312, 1200, 3280 or 7320 for creating a computer reconstruction containing less than 17 tiers. The corresponding nodes are marked red in Figure 5.17(a). Experimenting with the program `plantograph` shows that to create a reconstruction with regular bottom boundary we need to remove two nodes.

We remove the nodes as marked in Figure 5.20(a). The left node which we remove, is the place on the diagonal where four squares join. We have seen a similar situation by the reconstructions of the muqarnas in the north iwan of the Friday Mosque in Natanz (see Section 5.2.1) and the muqarnas in the east portal of the shrine of Bāyazīd in Bisṭām (see Section 5.2.2). In both cases we removed this node for creating the reconstruction which coincides with reality. The other node we remove is the place where six rhombi join. A similar situation can be found in the famous muqarnas vault over the tomb of Skaykh 'Abd aṣ-Ṣamad (see [Harb, 1978, pp. 55–58]). The muqarnas graph is now calculated by

```
plantograph -c -Qpl_tis_plate_new.fig.
```

and by setting the direction of the remaining undetermined edges so that the bottom boundary is regular, we calculate the computer reconstruction by running

```
graphtomuq -v1 -Ppl_tis_plate_new.fig -Gcrc_gr_tis_plate_new.fig.
```

Removing another set of nodes from the suggested ones results in other computer reconstructions. These reconstructions contain element combinations we have not seen in other muqarnas from the Il-Khanid period.

Yaghan proposed different computer reconstructions. We discuss here his interpretation with plane projection as given in Figure 5.21(a) and computer reconstruction as given in Figure 5.21(c). The corresponding muqarnas graph can be found by removing the nodes on the diagonal as marked in Figure 5.21(a). The graph is then calculated by

```
plantograph -c -Qrmlines_pl_tis_plate_yagh.fig,
```

And the corresponding computer reconstruction can be found by

```
graphtomuq -v1 -Ppl_tis_plate_yagh.fig -Ggr_tis_plate_yagh.fig.
(5.1)
```

The result is shown in Figure 5.21(c).

The diagonal nodes which need to be removed for creating Yaghan's reconstruction proposal are not suggested by the program `removelines`. The explanation for this can be found in the fact that Yaghan interprets the hexagons on the diagonal as cell with projection a square standing on two intermediate elements with plane projection rhombi. In the muqarnas plan this is not drawn: the hexagons are given by a jug, a square and two half-rhombi.

Yaghan has chosen to interpret most squares as an intermediate element with plane projection a large biped carrying a cell with plane projection a jug. Therefore, we need to

give the plane projection of Yaghan's interpretation in the input line (5.1). If the original design is used, cells with plane projection a square are used instead. The other proposed reconstructions of Yaghan contains uncommon element combinations and will therefore not be considered here (see [Yaghan, 2000] and [Dold-Samplonius and Harmsen, 2005]).

Comparing the different reconstructions corresponding to the design, we remark that the reconstruction of Harb contains 18 tiers, the reconstruction with regular center contains 17 tiers and Yaghan as our reconstruction contains 12 tiers. The main difference between the interpretation given by Harb (see Figure 5.18(c)) and our interpretation (see Figure 5.20(c)) is the global geometric form of the muqarnas. In Harb's interpretation the muqarnas starts in the four corners. The first tier consists only of two elements in these corners. The number of elements in subsequent tiers increases, which means that the length of the subsequent tiers grows until the elements span the whole circumference. This happens for the first time at the eighth tier. In our interpretation, and also in that of Yaghan, the elements span the whole circumference in each tier.

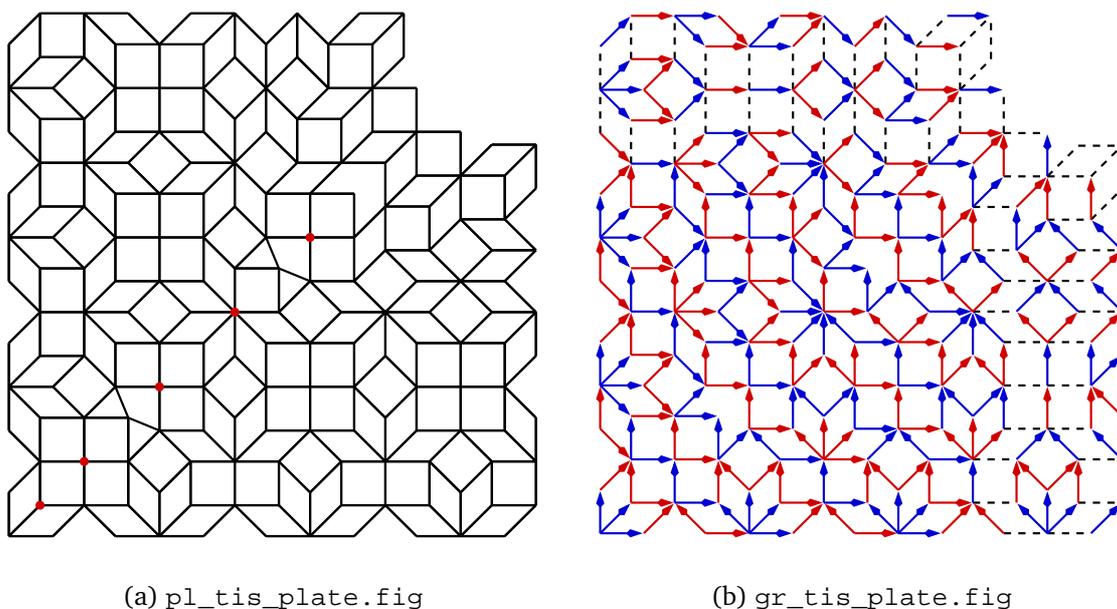
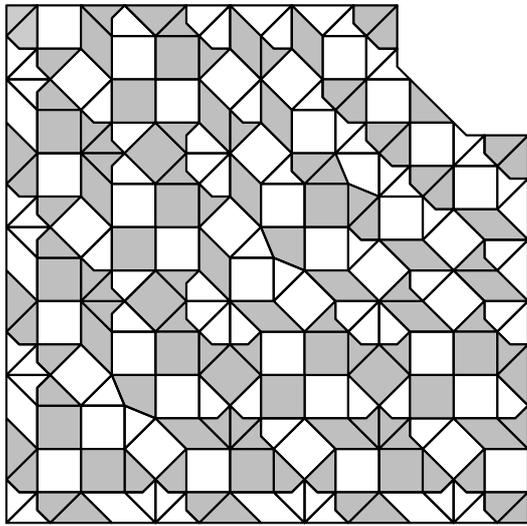
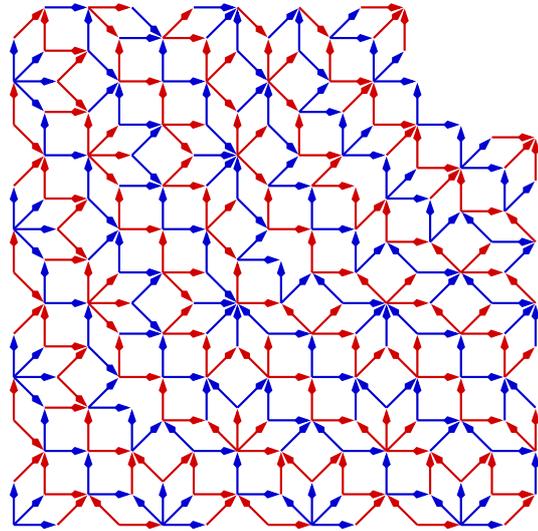


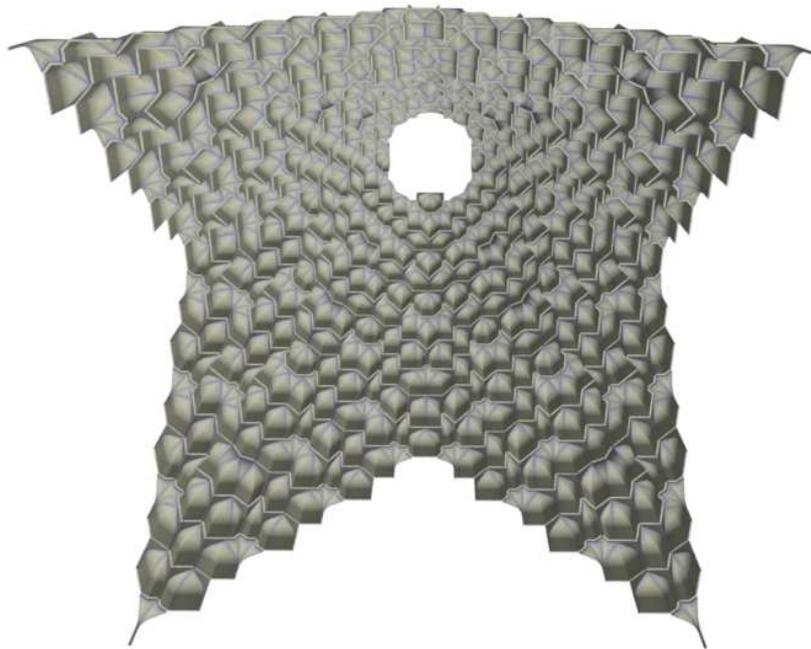
Figure 5.17. First steps in the analysis of the design found at the Takht-i-Sulaymān.



(a) Plane projection of Harb's interpretation of the design.



(b) gr_tis_plate_harb.fig



(c) WRL_harb/main.wrl

Figure 5.18. Reconstruction of Harb's interpretation of the design on the plate found at the Takht-i-Sulaymān.

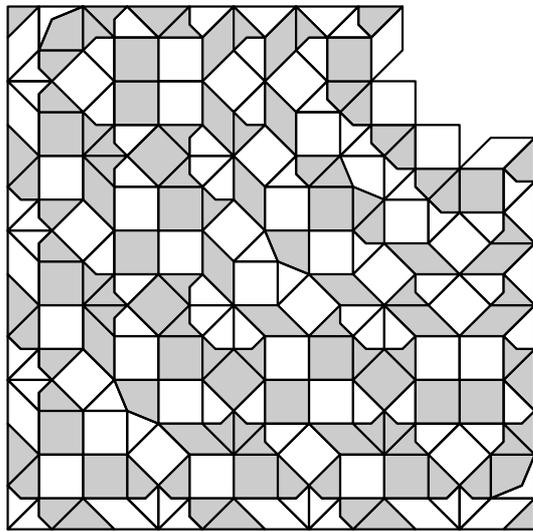
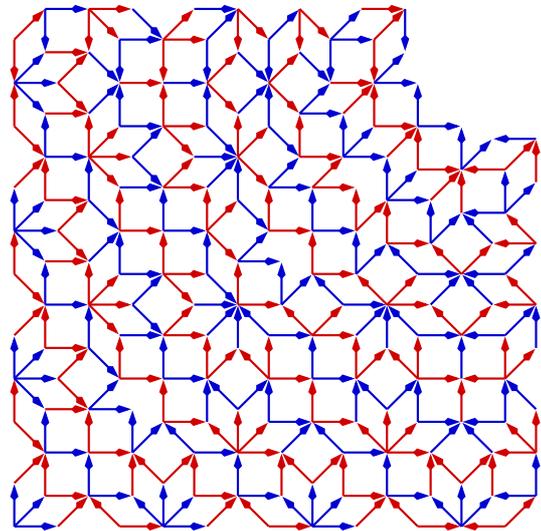
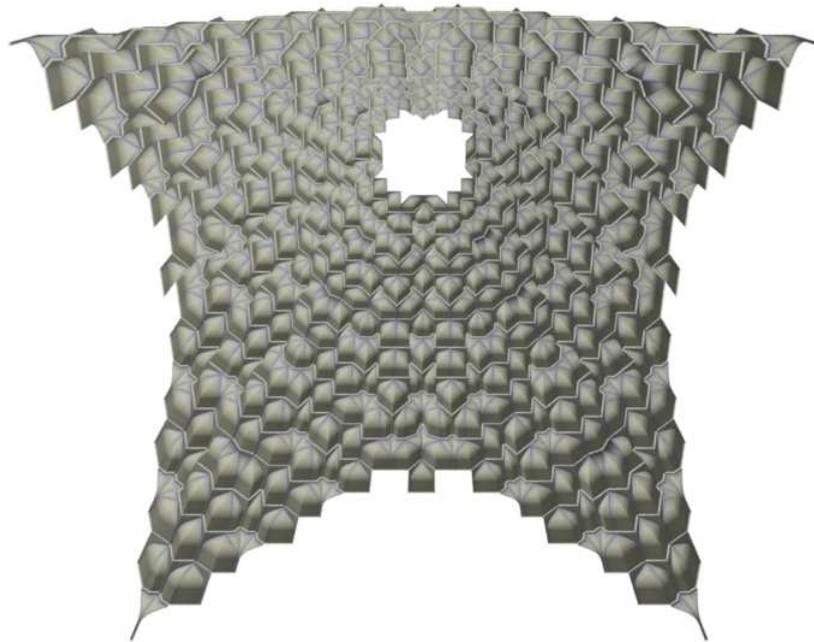
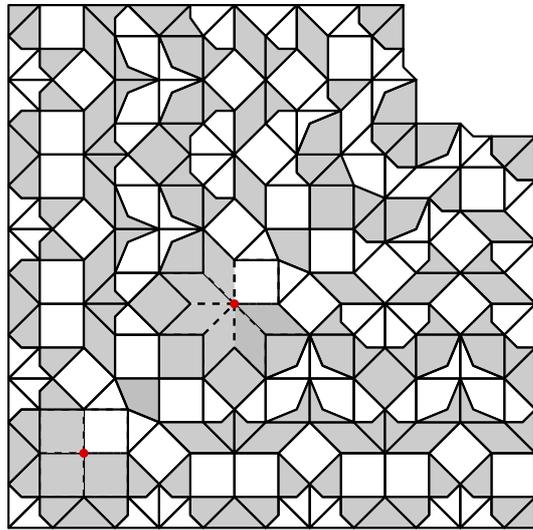
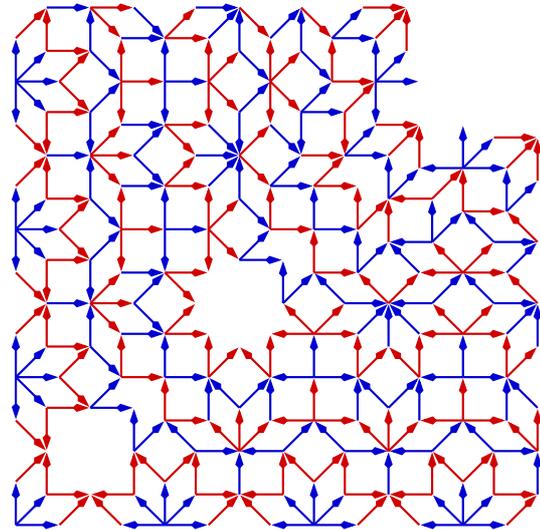
(a) `pl_tis_plate_regularcenter.fig`(b) `gr_tis_plate_regularcenter.fig`(c) `WRL_regularcenter/main.wrl`

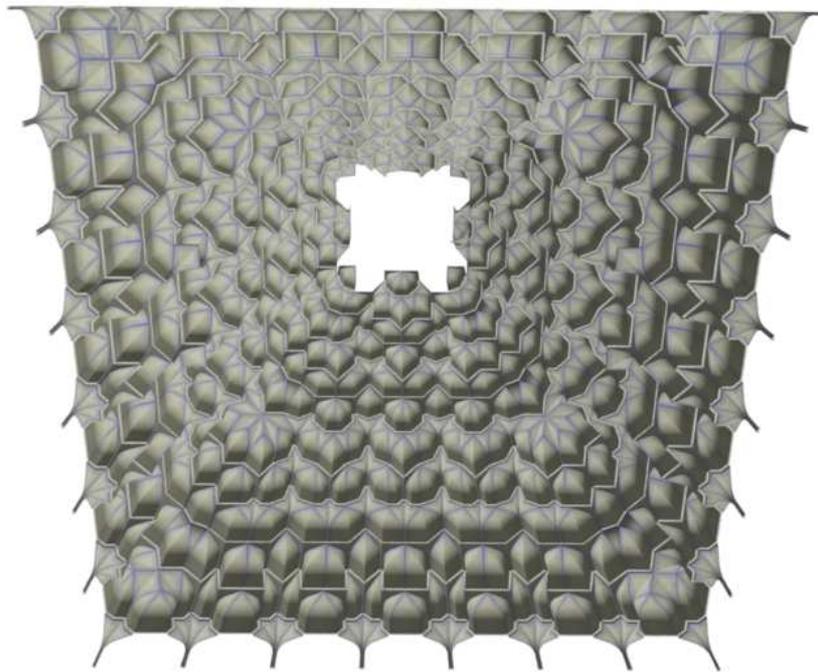
Figure 5.19. Reconstruction proposal of a muqarnas with regular center corresponding to the design on the plate.



(a) pl_tis_plate_our.fig

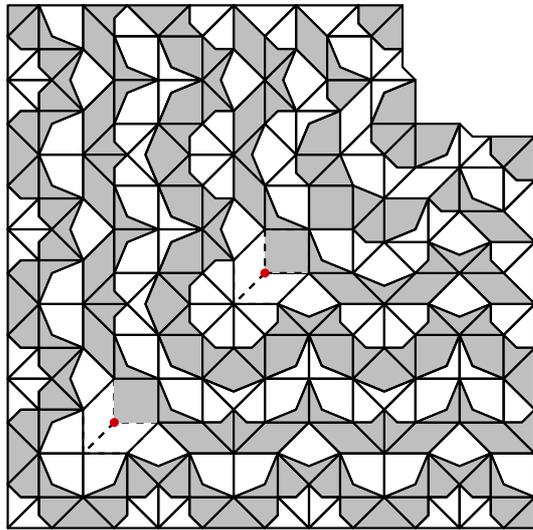


(b) gr_tis_plate_our.fig

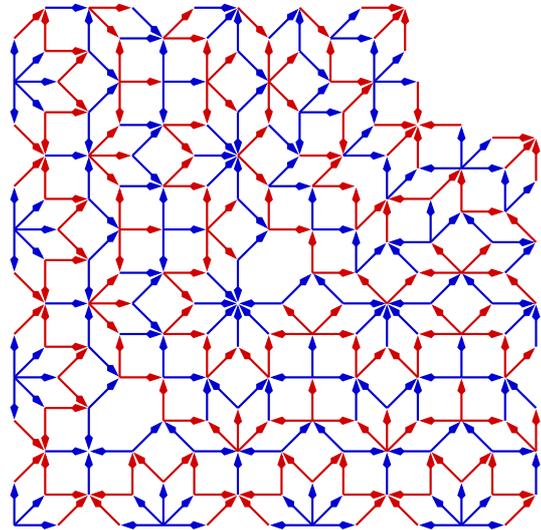


(c) WRL_our/main.wrl

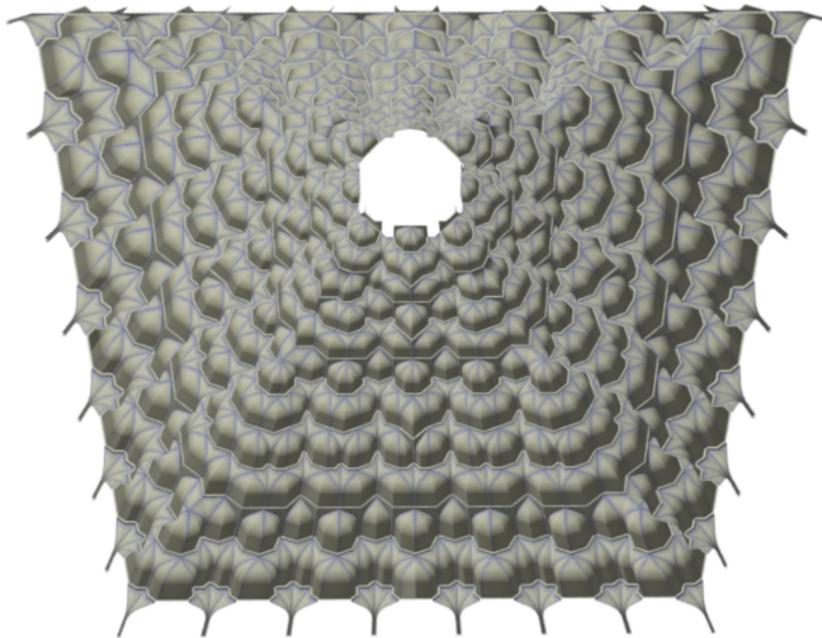
Figure 5.20. Reconstruction process if specified nodes on the diagonal are removed.



(a) pl_tis_plate_yagh.fig



(b) gr_tis_plate_yagh.fig



(c) WRL_yaghan/main.wrl

Figure 5.21. Reconstruction proposal of Yaghan.

Chapter 6

Conclusions

Reviewing our research questions, which we formulated in the introduction (see Section 1.2.1), we present the following answer:

There is an algorithm for reconstructing muqarnas. There are different muqarnas reconstructions with the same plane projection possible. The possibilities are reduced by the proportions of the vault into which the muqarnas should fit.

We started this work with a problem taken from the practice, concerning reconstructions of muqarnas, an architectonic decoration appearing in the Islamic architecture. By describing the problem in a more mathematical context, we created a framework in which we are able to analyze it. For this aim we defined in Chapter 2 the three-dimensional muqarnas structure and muqarnas plans, motivated by existing muqarnas vaults and properties of muqarnas designs. In Chapter 3 we have used this framework to analyze muqarnas from the Seljuk and Il-Khanid periods. We developed methods to study the structure of muqarnas directly from their designs. These methods were formulated so that it was possible to write software tools which are able to analyze muqarnas designs (see Chapter 4). In this context, we have written the computer programs `removelines`, `plantograph` and `graphtomuq`. From the examples of Chapter 5, we conclude that the created software is able to give a fast analysis of the designs and results in a better understanding of the structure of the muqarnas.

We defined the muqarnas structure by a set of muqarnas elements together with rules how they can be arranged (see Section 2.1). The combinations of elements covered by our definitions are those common in Seljuk and Il-Khanid architecture. A new way of representing the three-dimensional structure of the muqarnas in the design is given by drawing arrows. These arrows form a directed subgraph of the muqarnas design. Properties of the muqarnas structure lead to properties of this graph. This makes it possible to formulate different rules to find the graphs from the design with an allowed muqarnas realization. The software tool `plantograph` applies these rules to analyze a muqarnas design.

In the examples, taken from the Seljuk and Il-Khanid periods, we have seen that most directions are fixed by these rules and all possible muqarnas reconstructions cor-

responding to a given plan can be calculated. If we consider more general muqarnas, the properties can change and the rules need to be adapted. We may need to take into account more different element combinations and less directions in the graph can be set. We could state that the definitions depend on the style of muqarnas we want to reconstruct. Allowing other kind of element combinations also needs an adaptation of definitions. If the definitions cover more kind of element combinations, reconstruction can be more difficult.

With the program `graphtomuq` it is possible to create a muqarnas reconstruction directly from the graph. We have seen that there may be different muqarnas reconstructions possible with same plane projection. We distinguish locally and globally different muqarnas. Concerning globally different muqarnas we have seen that the direction of orbits not determined by the program `plantograph` may result in different interpretations with different shapes (see Section 5.2.3).

Another operation during the analysis, causing global different interpretations of a muqarnas plan is removing nodes on the diagonal of a plan. We have seen that by removing a node from the plan an island is constructed. In the three-dimensional reconstruction this island expands to a set of intermediate elements. By fixing locally the elements we use in the reconstruction, the whole muqarnas can be influenced. It is the task of the program `removelines` to select the edges of the plan which appear in the muqarnas graph and to find which nodes can be removed to create an island. If we know into which vault a muqarnas needs to fit, we know the height and therefore the number of tiers the muqarnas consists of. The number of possible muqarnas reconstructions corresponding to a plane projection can then be restricted (see Section 5.2.1).

Apart from globally different muqarnas, we also consider the locally different ones. Sometimes it is possible to exchange a muqarnas element by others without changing other parts of the muqarnas. The muqarnas graphs are then the same and this decision need to be considered during the three-dimensional reconstruction process, executed by the program `graphtomuq`.

We applied our algorithms on the oldest known muqarnas design, which is found at the Takht-i-Sulaymān. The last decennial different researchers studied the design and gave their interpretation of the design. This work shows how new methods give an overview of all possible interpretations of a design. By applying a more structural method of studying the muqarnas designs, we are able to propose suggestions differently from the interpretations found in the literature, but more realistic with the style of muqarnas from the same time and region.

Research about muqarnas could develop in different directions. One direction could be to extend our software tools so that they can handle other kind of muqarnas. Our algorithm is designed for muqarnas fitting into domes and niches, and therefore demands rectangular plane projections in the input. Muqarnas which appear as a decoration on minarets have a plane projection with another geometrical form. Those cannot be handled by our algorithm. Another definition of the boundary of a muqarnas plan is required to analyze such muqarnas. In other styles of muqarnas other elements and other combi-

nation rules may appear which influences the muqarnas structure.

As we now possess software tools to analyze muqarnas, it is possible to test new designed muqarnas on mistakes. For renovation purposes we can also first view the muqarnas reconstruction on the computer. If only a part of the muqarnas survived and therefore only a part of its plane projection is known, we need to fill the plane projection with polygons such that there is a muqarnas representation possible. This can be done by using studies about tilings to find the possibilities to fill the two-dimensional space with polygons. If the muqarnas consists only of basic elements, the muqarnas plan can be considered as an octagonal tiling (see e.g. [Senechal, 1995, p.213–218]). Studying what kind of assumptions are needed so that a finite part of a tiling has a muqarnas representation can be useful for reconstructing partly destroyed muqarnas or for constructing new muqarnas designs. Our software tools can then be used to validate whether there is a three-dimensional reconstruction possible.

We already created a data-structure based on integers in which we can represent tilings, given by the plane projection of muqarnas only consisting of basic elements. We could imagine that a better algebraic description of the structure of the plans is possible and useful. Our data-structure could then be more specified to exclude plans for which no muqarnas reconstruction is available.

An important initiation is made in order to study muqarnas in a more structural way. To have an algorithm that is able to analyze the structure of a muqarnas, leads to a better understanding of muqarnas designs. The integration of the computer in our research provides a fast analysis, which is an important advantage, compared with older methods based on trial and error.

Appendix A

Graph Theoretical Terminology

In this appendix we give an overview of the main definitions from graph theory, which we use in our research. The definitions are taken from [Lemmens and Springer, 1992, Chapter 7 and 10] and [Bollobás, 1979].

Definition A.1 (undirected graph). *An undirected graph $G = G(N, E)$ is an ordered pair of disjoint sets N, E such that E is a subset of the set of unordered pairs of N . The set N is the set of nodes and the set E is the set of edges.*

If the edge e is the unordered pair of nodes v and w , we call v and w the *end-nodes* of the edge e and write $e = \text{edge}(v, w)$. Because such an edge is defined as an *unordered* pair we identify $\text{edge}(v, w)$ with $\text{edge}(w, v)$. Often graphs are visualized by drawings. The nodes are represented in the plane as points, an edge $\text{edge}(v, w)$ is represented as a line connecting the points v and w (see Figure A.1).

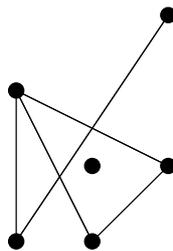


Figure A.1. Visual representation of an undirected graph.

We create a directed graph by giving each edge a direction. Then the pairs $\{v, w\}$ are ordered pairs and we use the word *arrow* to refer to the directed edges. The direction is in its visual representation given by an arrow pointing from v to w .

Definition A.2 (directed graph). *A directed graph $G = G(N, C)$ is an ordered pair of disjoint sets N, C such that C is a subset of the set of ordered pairs of N . The set N is the set of nodes and the set C is the set of arrows.*

We say that an arrow c given by the ordered set $\{v, w\}$ points from v to w , we write $c = \text{arw}(v, w)$. The node v is called the *initial point* of the arrow c and the node w is called its *end-point*. Both v and w are called *end-nodes* of the arrow c . We denote c_s for the initial point of an arrow c and c_t for its end-point.

Definition A.3 (subgraph). A graph $H(M, F)$ is a subgraph of $G(N, E)$ if $M \subset N$ and $F \subset E$. We write $H \subset G$ for the subgraph H of G .

The subgraph of $G(N, E)$ induced by the set of nodes $M \subset N$ is defined by the subgraph $H(M, F)$ such that $\text{edge}(v, w) \in E$ belongs to F if and only if both end-nodes $v, w \in M$.

To refer to parts in the graph connected to nodes or edges, we use the terms *adjacent* or *neighbor* and *incident*. In the case of a directed graph, a neighbor of a node can be a *predecessor* or a *successor*, depending on the direction of the arrows incident to these nodes.

Definition A.4 (adjacent, neighbor). We say that two nodes $v, w \in N$ of a graph $G(N, E)$ are adjacent or neighbors, if there is an edge $e \in E$ with end-nodes v and w .

Definition A.5 (incident). Two edges e, f are incident if they have exactly one common end-node. An edge e is incident to node v if v is an end-node of e .

Definition A.6 (predecessor). Let $G = G(N, C)$ be a directed graph and $v, w \in N$. We say that node w is a predecessor of v if there is an arrow $c \in C$ pointing from w to v .

Definition A.7 (successor). Let $G = G(N, C)$ a directed graph and $v, w \in N$. We say that node w is a successor of v if there is an arrow $c \in C$ pointing from v to w .

The notion of *path* is used to denote connections in the graph. The *length* of a path indicates distances between nodes. If the initial point and the end-node of a path are the same, we call this path a *circuit*. The circuits which are of most interest of us, are those which do not coincide themselves, the so called *cycles*.

Definition A.8 (path). A path p of length $l(p) = n$ in an undirected graph $G(N, E)$ is an ordered sequence of nodes $v_1, \dots, v_n \in N$, such that there exist edges $\text{edge}(v_i, v_{i+1})$ in E for $i = 1, \dots, n - 1$. We write $p_s = v_1$ for the initial point of the path p and $p_t = v_n$ for the end-point of p . Similar, an ordered sequence of nodes $v_1, \dots, v_n \in N$ defines a path p of length $l(p) = n$ in a directed graph $G(N, C)$ if there exist arrows $\text{arw}(v_i, v_{i+1})$ in C for $i = 1, \dots, n - 1$.

Definition A.9 (connected graph). An undirected graph G is connected if for all nodes v, w there is a path with initial point v and end-point w . A directed graph G is connected if its underlying undirected graph is connected. This means that for all nodes v, w there is a sequence $v_1, \dots, v_n \in C$ with $v_1 = v$ and $v_n = w$ such that $\text{arw}(v_i, v_{i+1}) \in C$ or $\text{arw}(v_{i+1}, v_i) \in C$ for $i = 1, \dots, n - 1$.

Definition A.10 (cycle). A cycle c in a graph is a path $c = v_1, \dots, v_n$, such that $v_1 = v_n$ and $v_i \neq v_j$ for $i, j = 1, \dots, n - 1, j \neq i$.

Index

- adjacent, 118
- Alay Han, 84
- almond, 14
- angle, 19
- Ankara, Arslanhane Camii, 92
- apex, 13
- arrow, 117
- backside, 13
- barley kernel, 16
- biped
 - large, 13
 - small, 14
- Biṣṭām, Shrine of Bāyazīd, 98
- bottom, 18
- bottom boundary, 25
- boundary, 25
- cell, 12
- center
 - muqarnas, 19
 - plan, 25
- central node, 19
- circuit, 118
- connected
 - graph, 118
- curved side, 13
- cycle, 118
- diameter, 19
- edge, 117
- element
 - position, 21
 - type, 20
- end–node
 - arrow, 118
 - edge, 117
 - end–point, 118
- face, 45
 - cell–, 45
 - int–, 45
- Farūmad, Friday Mosque, 100
- figure, 23
- global different, 55
- graph
 - directed, 117
 - muqarnas, 32
 - undirected, 117
- incident, 118
- induced, 118
- initial point, 118
- intermediate element, 12
- island, 23
- jug, 13
- Kayseri
 - Çifte Medrese, 86
 - Sultan Han, 89
- local different, 55
- module, 13
- Natanz, Friday Mosque, 95
- neighbor, 118
- node, 117
- non–singular, 41
- opposite, 37
- orbit, 39

- parallel, 39
- path, 118
- plan, 22
 - simplified, 49
- regular
 - bottom, 18
 - bottom boundary, 26
 - center, 19, 26
- rhombus, 13
- separate, 41
- singular, 41
- sink, 43
- size, 69
- source, 43
- square, 13
- standard position, 20
- subgraph, 118
- Takht-i-Sulaymān
 - Plate, 105
 - South Octagon, 103
- tier, 17
- Topkapı Scroll, 5

Acknowledgements

This thesis never could have been realized without the help of different people. I am grateful to Professor G. Reinelt who gave me the possibility to write this thesis about my research on muqarnas. This study on muqarnas started with the project “Mathematische Grundlagen und computergraphische Rekonstruktion von Stalaktitengewölben (Muqarnas) in der Islamischen Architektur”, financed by the German *Bundesministerium für Bildung und Forschung (BMBF)* under Grant 03WNX2HD. I am deeply indebted to Yvonne Dold-Samplonius, for initiating this project and sharing her experience with me. I would like to thank Professor W. Jäger, who supported the initiative of this project, and Professor H. G. Bock, who enabled me to finish it.

Professor Jan Hogendijk motivated me to start the project and was very helpful during the process. I would like to thank him for that and for his faith in me. I also want to thank Susanne Krömker and Michael Winckler for coaching me during my research and giving me the possibility to work in the *Computer Graphics* group, where I had the possibility to develop my knowledge in different areas.

Bernhard Keil and Daniel Jungblut assisted with a lot of work. In particular, I want to mention the preparation of the input plans, so that my algorithms were able to read them for creating reconstructions. They also made it possible to convert the VRML-data into other data-formats, which enabled them to revise the reconstructions with different programs. Beautiful animations were one of the results. Daniel has written a user interface `planedit` which makes it much easier to work with the algorithms. Bernhard supported in installing and testing programs for me. I thank them both for their assistance and interest in my work.

I am grateful to Professor Shiro Takahashi for sharing data from his large database of muqarnas plans. By being allowed to use his plans, I saved a lot of time. I also would like to thank Mohammed Yaghan for his visit to Heidelberg for discussing about muqarnas with me.

I want to acknowledge the people who gave me the permission to use their pictures in this thesis. Pictures not made by myself stem from Susanne Krömker, Mohammed Bagheri, Yvonne Dold-Samplonius and Michael Winckler.

Finally, I would like to thank my family, friends and colleagues who made an enjoying stay in Heidelberg possible. In particular, I want to thank Mariya, Pau and Frank for listening and discussing, not only about my thesis. Frank, I want to thank you for your many useful suggestions and your non-exhausting patience during the preparation of this thesis.

Bibliography

- [al Asad, 1994] al Asad, M. (1994). Applications of Geometry. In Frishman, M. and Khan, H.-U., editors, *The Mosque. History, Architectural Development & Regional Diversity*, pages 55–70. Thames and Hudson, London.
- [al Kāshī, 1558] al Kāshī, G. a.-D. n. (1427/1558). *Miftāḥ al-Ḥisāb (Key of Arithmetic)*. Ms. Or. 185, Leiden.
- [Ames et al., 1997] Ames, A. L., Nadeau, D. R., and Moreland, J. L. (1997). *VRML 2.0 Sourcebook*. John Wiley & Sons Inc, New York.
- [Aslanapa, 1971] Aslanapa, O. (1971). *Turkish Art and Architecture*. Faber and Faber Limited, London.
- [Bianca, 2000] Bianca, S. (2000). *Urban Form in the Arab World*. Hochschulverlag AG an der ETH Zürich, Institut für Orts-, Regional- und Landesplanung (ORL-Institut), ETH Zürich.
- [Bollobás, 1979] Bollobás, B. (1979). *Graph Theory, an Introductory Course*. Springer-Verlag, New York.
- [Castéra, 1996] Castéra, J.-M. (1996). *Arabesque, Art décoratif au Maroc*. ACR Édition Internationale, Paris.
- [Dold-Samplonius, 1992] Dold-Samplonius, Y. (1992). *Practical Arabic Mathematics: Measuring the Muqarnas by al-Kāshī*, volume 35 of *Centaurus*, pages 193–242. Munksgaard.
- [Dold-Samplonius, 1996] Dold-Samplonius, Y. (1996). How al-Kāshī Measures the Muqarnas: A Second Look. *Mathematische Probleme im Mittelalter: Der lateinische und arabische Sprachbereich*, Wolfenbütteler Mittelalter-Studien Vol. 10:56–90.
- [Dold-Samplonius and Harmsen, 2004] Dold-Samplonius, Y. and Harmsen, S. (2004). Muqarnas, Construction and Reconstruction. In Williams, K. and Cepeda, F. D., editors, *Nexus V: Architecture and Mathematics*, pages 69–77, Fucecchio (Florence). Kim Williams Books, 2004.

- [Dold-Samplonius and Harmsen, 2005] Dold-Samplonius, Y. and Harmsen, S. L. (2005). The Muqarnas Plate found at Takht-i Sulayman: A New Interpretation. *Muqarnas, An Annual on the Visual Culture of the Islamic World*, 22:82–91.
- [Dold-Samplonius et al., 2002] Dold-Samplonius, Y., Harmsen, S. L., Krömker, S., and Winckler, M. J. (2002). Magic of Muqarnas, Video about Muqarnas in the Islamic World. IWR-Preprint 2002-39.
- [Erdmann, 1961] Erdmann, K. (1961). *Das anatolische Karavansaray des 13. Jahrhunderts*. Verlag gebr. Mann.
- [Foley et al., 1994] Foley, J. D., van Dam, A., Feiner, S. K., Hughes, J. F., and Phillips, R. L. (1994). *Grundlagen der Computergraphik*. Addison-Wesley.
- [Grabar, 1992] Grabar, O. (1992). *The Alhambra*. Harvard University Press.
- [Harary, 1974] Harary, F. (1974). *Graphentheorie*. R. Oldenbourg Verlag, München.
- [Harb, 1978] Harb, U. (1978). *Ilkhanidische Stalaktitengewölbe, Beiträge zu Entwurf und Bautechnik*, volume IV of *Archäologische Mitteilungen aus Iran*. Dietrich Reiner, Berlin.
- [Herzfeld, 1942] Herzfeld, E. (1942). Damascus: Studies in Architecture. *Ars Islamica, The Research Seminary in Islamic Art*, IX:1–52. University of Michigan.
- [Ibrahim, 2002] Ibrahim, S. M. A. (2002). *Muqarnas, ein Stalaktitenförmiges Architekturelement - sein Struktureller Aufbau und seine Bildungsregeln*. PhD thesis, Universität Stuttgart.
- [Jones and Goury, 2001] Jones, O. and Goury, J. (2001). *Planos, Alzados, Secciones y Detalles de la Alhambra*. María Ángeles Campos Romero.
- [Jungblut, 2005] Jungblut, D. (2005). Muqarnas Plan Editor. <http://pille.iwr.uni-heidelberg.de/~muqtool/>. Last Visit: 19th November 2006.
- [Lemmens and Springer, 1992] Lemmens, P. and Springer, T. (1992). *Hoofdstukken uit de Combinatoriek*. Epsilon Uitgaven, Utrecht.
- [Necipoğlu, 1995] Necipoğlu, G. (1995). *The Topkapi Scroll – Geometry and Ornament in Islamic Architecture*, Topkapi Palace Library MS. H. 1956. The Getty Center, Santa Monica.
- [Noltemeier, 1975] Noltemeier, H. (1975). *Graphentheorie mit Algorithmen und Anwendungen*. Walter de Gruyter & Co., Berlin.
- [Notkin, 1995] Notkin, I. (1995). Decoding Sixteenth-Century Muqarnas Drawings. *Muqarnas, An Annual on the Visual Culture of the Islamic World*, 12:148–171.

- [Pope, 1939] Pope, A. U., editor (1939). *A Survey of Persian Art, from prehistoric times to the present*, volume II. Oxford University Press, London and New York.
- [Sato and Smith, 2002] Sato, T. and Smith, B. V. (2002). Xfig User Manual. <http://www-epb.lbl.gov/xfig/>. Last Visit: 19th November 2006.
- [Sedgewick, 2002] Sedgewick, R. (2002). *Algorithms in C++, Graph Algorithms*. Addison-Wesley, third edition.
- [Senechal, 1995] Senechal, M. (1995). *Quasicrystals and Geometry*. Cambridge University Press, Cambridge.
- [Takahashi, 2004] Takahashi, S. (2004). <http://www.tamabi.ac.jp/idd/shiro/muqarnas/>. Last Visit: 19th November 2006.
- [Wilber, 1955] Wilber, D. N. (1955). *The Architecture of Islamic Iran, The Il Khānid Period*. Princeton University Press, Princeton New Jersey.
- [Yaghan, 2000] Yaghan, M. A. J. (2000). Decoding the two dimensional Pattern found at Takht-i Sulayman into three-dimensional Muqarnas Forms. *Iran*, XXXVIII:77–95.
- [Yaghan, 2001a] Yaghan, M. A. J. (2001a). *The Islamic Architectural Element “Muqarnas”*. Phoibos Verlag, Vienna.
- [Yaghan, 2001b] Yaghan, M. A. J. (2001b). The Muqarnas Pre-designed Erecting Units: Analysis, Definition of the Generic Set of Units, and a System of Unit-Creation as a New Evolutionary Step. *Architectural Science Review*, 44(3):297–309.
- [Yaghan, 2003] Yaghan, M. A. J. (2003). Gadrooned-Dome’s Muqarnas-Corbel: Analysis and Decoding Historical Drawings. *Architectural Science Review*, 46(1):69–88.