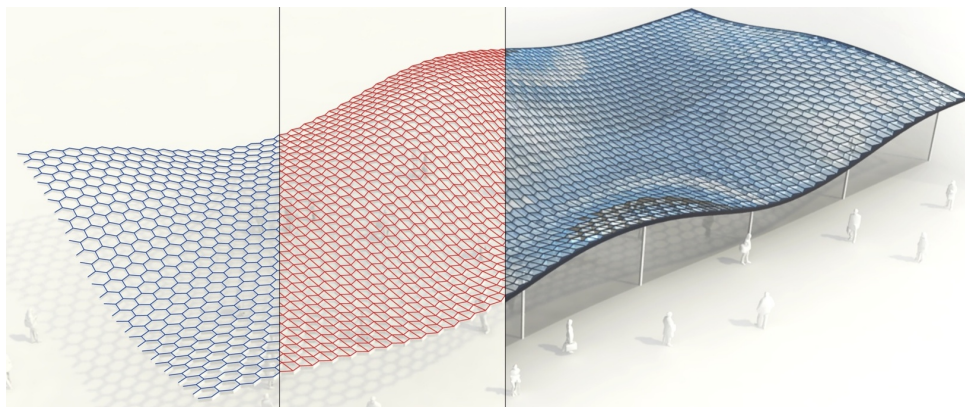


# Freeform Honeycomb Structures

Caigui Jiang<sup>1</sup> Jun Wang<sup>1</sup> Johannes Wallner<sup>2</sup> Helmut Pottmann<sup>1,3</sup>

<sup>1</sup> KAUST <sup>2</sup> TU Graz <sup>3</sup> TU Wien



**Figure 1:** We approximate the Cour Visconti roof in the Louvre, Paris, by a quad mesh with planar faces which caps a honeycomb (consisting of hexagonal cells whose walls intersect at  $120^\circ$ ). This structure exhibits several features important in freeform architectural design: planar faces, low valence of nodes, a torsion-free support structure, and repetitive node geometry.

---

## Abstract

Motivated by requirements of freeform architecture, and inspired by the geometry of hexagonal combs in beehives, this paper addresses torsion-free structures aligned with hexagonal meshes. Since repetitive geometry is a very important contribution to the reduction of production costs, we study in detail “honeycomb structures”, which are defined as torsion-free structures where the walls of cells meet at  $120$  degrees. Interestingly, the Gauss-Bonnet theorem is useful in deriving information on the global distribution of node axes in such honeycombs. This paper discusses the computation and modeling of honeycomb structures as well as applications, e.g. for shading systems, or for quad meshing. We consider this paper as a contribution to the wider topic of freeform patterns, polyhedral or otherwise. Such patterns require new approaches on the technical level, e.g. in the treatment of smoothness, but they also extend our view of what constitutes aesthetic freeform geometry.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms

---

## 1. Introduction

Nature’s design strategies and solutions are a rich source of inspiration for various branches of science and technology (see e.g. the biomimicry web pages [www.asknature.org](http://www.asknature.org) and [biomimicry.net](http://biomimicry.net)). Architecture and structural engineering are certainly among those areas which learn from nature, and this should be especially true for research on the realiza-

tion of complex architectural structures. The present paper presents a contribution in this direction. It is inspired by *honeycombs* which possess fascinating freeform shapes (see Figure 1) and are composed of hexagonal cells whose faces meet at angles close to  $120$  degrees. Structures containing honeycomb geometry have been used in engineering for a long time, as a means to minimize material without losing structural strength [LSZ\*14]. This paper assumes a differ-



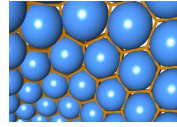
**Figure 2:** Left: Natural honeycomb. Right: Design by E. van Egeraat based on a hexagonal torsion-free support structure.

ent viewpoint and sees them in the context of architectural geometry, as *torsion-free support structures* with congruent regular nodes and hexagonal cells. The most important property of such honeycomb structures is that all nodes are congruent within a reasonable tolerance, which should facilitate the fabrication of these structures. We will show how to compute and design honeycomb structures, discuss applications and their limitations, and we also discuss the new topic of *polyhedral patterns*.

**Related Work.** Given a base mesh  $M$  of any connectivity, computing a torsion-free structure based on  $M$  requires us to assign a plane to each edge so that the planes around each vertex (*node*) intersect in a straight line, which is called the *node axis*. In the actual realization of this structure, beams are positioned symmetric to the edge planes. For triangle meshes, torsion-free support structures have very few degrees of freedom and are not interesting for applications. This is because all node axes need to pass through a fixed point, or are parallel [PLW\*07]. For meshes with planar faces, torsion-free support structures have been well studied in recent years: They are accessible via the concept of *parallel meshes*. For quad meshes, support structures are related to *discrete line congruences* [WJB\*13]. Neither applies to honeycomb structures, which are based on hex meshes with non-planar faces.

Special hex-dominant torsion-free structures can be derived from triangle meshes with the circle-packing property [SHWP09] which are equipped with a packing of spheres centered in the vertices. The sphere's tangent planes in the points of contact form a torsion-free hex structure, and if all spheres are of the same size it will be a honeycomb with 120 degree intersection angles. Such CP meshes therefore are useful to initialize some optimization tasks discussed in this paper. A more general construction of support structures has been briefly addressed in connection with cell packing structures [PJH\*14], but without aiming at congruent nodes.

*Repetition of elements* has been a successful ingredient in reducing the fabrication cost of freeform architecture, see e.g. [SEKC13], and thus this topic already received some attention. Singh and Schaefer [SS10] optimize triangle meshes so that there is only a relatively small set of different faces



**Figure 3:** Hexagonal support structure derived from a circle-packing mesh, each cell containing a sphere touching the walls [SHWP09].

up to some chosen tolerance. Similarly, Fu et al. [FLHCO10] reduce the number of essentially different faces in non-polyhedral quad meshes. For arbitrary freeform shapes, the goal of congruent faces appears to conflict mesh fairness. We expect a similar effect for the goal of congruent nodes in meshes with straight edges. However, giving up the straightness of edges and using circular arcs instead, one can achieve congruent and even regular nodes for triangular, quad and hex combinatorics [BPK\*11].

Last, but not least we point to [ZCBK12], where the topics of repetition and general polyhedral patterns are combined by the analysis of the realization of surfaces with a bounded number of “folding elements”.

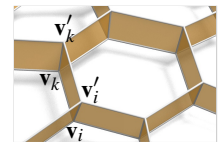
## Contributions

1. We analyze the geometry of honeycomb structures and discuss their flexibility to approximate freeform shapes. It turns out that a honeycomb can be orthogonal to a given surface only if that surface is developable. The Gaussian image of a honeycomb is an unusual kind of spherical 2D tiling which is concentrated on curves.
2. We present computation and interactive design of honeycombs, addressing special cases and particular applications relevant to architectural design.
3. Honeycombs can be converted to polyhedral patterns on surfaces, e.g. quad patterns consisting of planar faces. Such patterns, which are *not* a discrete version of a smooth curve network, constitute a remarkable and novel topic.

## 2. Geometry and flexibility of honeycomb structures

This section discusses geometric properties of honeycombs. Our conclusions in particular yield insights in the degrees of freedom which are available when approximating a design surface by a honeycomb structure. It turns out that the condition of *congruent nodes* is rather strong: It implies that the honeycomb can always follow a given surface, but in general it is not possible that the node axes remain orthogonal to it. We show that this can happen only for developable surfaces.

**Definitions.** We summarize again the definition of a honeycomb structure: It requires two combinatorially equivalent meshes  $M = (V, E, F)$  and  $M' = (V', E', F')$  such that each pair  $\mathbf{v}_i \mathbf{v}_k, \mathbf{v}'_i \mathbf{v}'_k$  of corresponding edges defines a planar quadrilateral *wall* which serves to border the open cells associated with the faces of the mesh. Further we require that for each



vertex the incident walls form angles of 120 degrees (implying that the valence of each vertex is  $\leq 3$  and faces will be mostly hexagonal). The intersection of walls at a vertex  $v_i$  is called its *node axis* (it is the straight line  $v_i v_i'$ ).

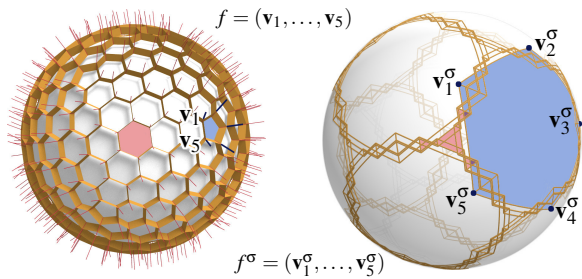
The most relevant geometric information contained in a honeycomb structure is not the meshes  $M, M'$  but the edge planes which carry the walls (connecting corresponding edges), and the node axes which connect corresponding vertices. If these data are given we may freely choose vertices of  $M, M'$  in the node axes, thereby defining a honeycomb structure (this e.g. implies that any surface can trivially be approximated by a honeycomb structure, by moving the vertices of an existing honeycomb close to that surface).

We will see that for our computations it is mostly sufficient to encode the relevant geometric data of honeycombs by the vertices of the mesh  $M$ , and the normal vectors of walls in the honeycomb.

**The spherical image of a honeycomb.** In order to understand the global geometry of honeycombs we study the unit vectors  $v_i^\sigma$  which indicate the directions of the node axis  $v_i v_i'$ . In our applications the base mesh is always following a smooth surface, and we assume that the vectors  $v_i^\sigma$  consistently point to one side of that surface. For any face  $f = (v_{i_1}, v_{i_2}, \dots)$  of the base mesh, the *spherical image* of that face, resp. cell, is the spherical polygon  $f^\sigma = (v_{i_1}^\sigma, v_{i_2}^\sigma, \dots)$ , see Figure 4. The spherical image  $M^\sigma$  of the honeycomb consists of the spherical images of the individual cells.

*Remark:* There is a linear space of honeycomb structures which belong to a given spherical image. The dimension of that space is  $|E| - |V| \approx \frac{1}{2}|V|$ , because there is one degree of freedom per edge plane, and one condition per vertex.

**The spherical zero-area property of honeycombs.** Since the walls of a honeycomb cell intersect at 120 degrees, the spherical image  $f^\sigma$  of this cell is a polygon whose edges are great circles intersecting at  $60^\circ$  or  $120^\circ$ . Figure 4 exhibits two kinds of cells:  $f^\sigma$  may be large (like the highlighted 5-



**Figure 4:** Honeycomb and its spherical image. Left: Honeycomb with node axes. A 6-gon and a 5-gon are highlighted in red and blue, resp. Right: Spherical image  $M^\sigma$  of node axes, with analogous highlighting. Observe the decomposition of  $M^\sigma$  into curve-like structures.

gon), or it might be small, meaning that the node axes of the corresponding cell point in roughly the same direction.

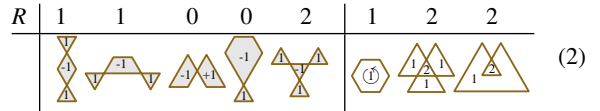
Recall the Gauss-Bonnet formula which relates the total rotation number  $R$  of a geodesic polygon in a surface with the Gauss curvature  $K$  and the angles  $\alpha_i$  which indicate the turning of the edge in the individual vertices. We have

$$\sum \alpha_i = 2\pi R - \int_x K(x)w(x), \quad (1)$$

where  $w(x)$  is the number of times the polygon is winding around the point  $x$ .

*Example:* For the blue polygon  $f^\sigma$  in Figure 4 we have  $K = 1$ , all  $\alpha_i$  equal  $60^\circ$ ,  $R = 1$ , and the winding coefficient  $w(x)$  equals 1 inside the polygon, and 0 outside. Thus  $\int K(x)w(x)$  reduces to the area of the polygon, and we get  $5 \times \frac{\pi}{3} = 2\pi - \text{area}(f^\sigma) \implies \text{area}(f^\sigma) = \frac{\pi}{3}$ .

We sketch some planar 6-gons whose angles are  $60^\circ$  or  $120^\circ$ , indicating  $R$  and the local winding number  $w(x)$ :



The turning angles  $\alpha_i$  in vertices obey  $\sum \alpha_i = 2\pi R$ , which follows from (1) when we let  $K = 0$ . A spherical 6-gon  $f^\sigma$ , on the other hand, satisfies Gauss-Bonnet with  $K = 1$ :

$$\sum \alpha_i = 2\pi R - \int_x w(x) = 2\pi R - \text{area}(f^\sigma), \quad (3)$$

where  $\text{area}(f^\sigma)$  is the oriented area of  $f^\sigma$ , defined as  $\int w(x)$ , meaning that each point contributes to the area with a certain multiplicity defined by how often the polygon winds around that point. If  $f^\sigma$  has the same combinatorics and angles as a planar hexagon, then  $\sum \alpha_i = 2\pi R$  immediately implies

$$\text{area}(f^\sigma) = 0. \quad (4)$$

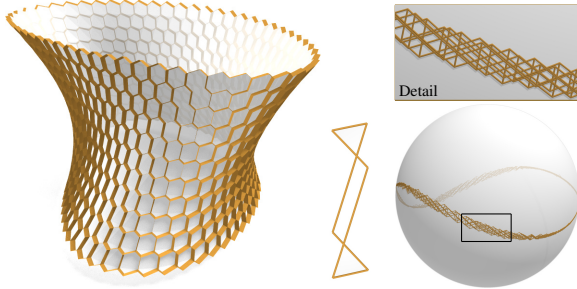
There may be  $f^\sigma$ 's whose area does not vanish,  $\sum \alpha_i \neq 2\pi R$ , and we cannot find a planar 6-gon with the same angles and rotation number  $R$ . However the smallest nonzero value of  $|\text{area}(f^\sigma)|$  equals  $\pi/3$  because both  $\sum \alpha_i$  and  $2\pi R$  are integer multiples of that number (which, incidentally, is the area of the highlighted 5-gon in Figure 4). We have proved:

**Proposition 1.** *The oriented area of a spherical image  $f^\sigma$  of a honeycomb cell either vanishes or has absolute value  $\geq \pi/3$ . In the first case there is a planar hexagon having the same angles  $\alpha_i$  and rotation number  $R$  as  $f^\sigma$  does.*

The practical implication of this statement is that a 6-gon honeycomb cell whose node axes are close together will have a spherical image  $f^\sigma$  of zero area. Figure 4 shows an example of this: All hexagons in the spherical image have zero area, and their shapes correspond to the examples in the left hand section of Equ. (2).

**Global distribution of node axes.** Numerical evidence (see e.g. Figures 4 and 5) shows that the spherical image of a

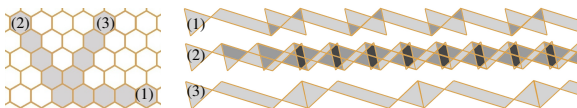
honeycomb exhibits curve-like structures where zero-area hexagons cluster, interspersed with individual polygons of nonzero area (non-hexagonal ones mainly, but also hexagons are possible). We argue why *smoothness* of honeycombs is responsible for this behaviour.



**Figure 5:** This curve-like spherical image (right) of a honeycomb (left) has the aspect of a tiling of the sphere by zero-area hexagons shaped like the one shown above.

This smoothness is not meant in the literal sense, since the shape of spherical image polygons entails high-frequency oscillation of node axes in any case. Instead we require a very modest kind of regularity: In a “regular” area of the honeycomb, the spherical images of two neighbouring cells should have roughly the same size and shape. The tiling of cells of the honeycomb must translate to a tiling of hexagons on the unit sphere. We therefore have to ask which of the hexagons shown in Equation (2) can possibly have a similar-shaped hexagon as a neighbour, along each of its six edges.

For polygons of type  $\bowtie$  (also depicted in Figure 5), this is possible. There is a combinatorially regular two-dimensional tiling of such hexagons which is difficult to visualize, but which can be seen in Figure 5. Three individual rows of hexagons contained in such a tiling are shown below, at right. For comparison we also show three rows of hexagons in the tiling with a geometrically regular hexagon (at left).



Geometrically, this tiling exists because the sum  $\alpha_1 + \alpha_3 + \alpha_5$  of turning angles is an integer multiple of  $360^\circ$ . For the polygons  $\nabla$ ,  $\diamond$ ,  $\triangle$ , such a tiling does not exist, because if they are made to have zero area, there is at least one pair of opposite edges of very different lengths. For polygons of type  $\triangle$ , opposite edges do have the same length, but an attempt at tiling will still fail, because the above-mentioned angle condition is not satisfied.

The image of the three rows also shows that a tiling of hexagons of type  $\bowtie$  is covering only a curve-like strip (see also Fig 5). For the planar version of this hexagon this is because the 6 parallel translations which map an edge to

its opposite edge all go in the same direction; for the spherical version of this hexagon this property holds only approximately. We summarize:

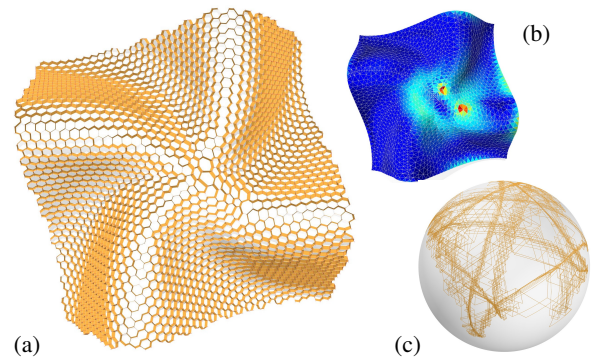
**Property 2.** *The spherical image of a regular honeycomb is curve-like.*

The practical implication of this statement is that node axes of a honeycomb will not behave like the normals of a surface, even if high-frequency oscillations are discarded. Only for *developable* surfaces, the spherical image of normals is curve-like (see Figure 6). This implies:

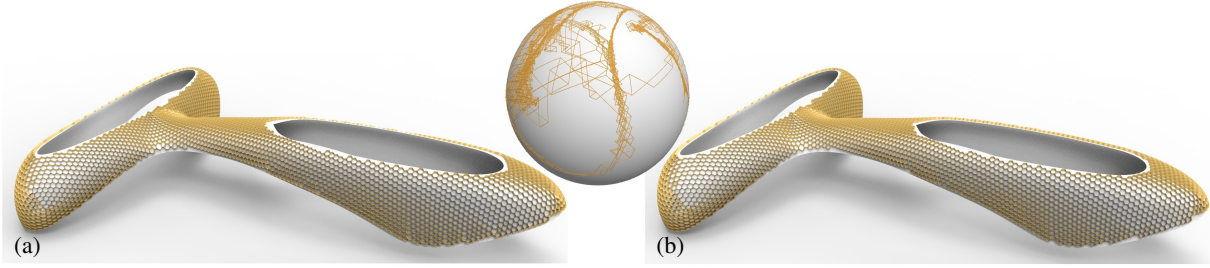
**Property 3.** *Node axes of a regular honeycomb approximate the normals of a surface only if that surface is developable.*

*Remark:* We also ask the converse question: Can we find a “zero-area” tiling along a given spherical curve  $C$ , which is the spherical image of a honeycomb structure? The answer is yes by the following heuristic argument: we start with a zigzag polyline following  $C$  and extend it by adding zero-area hexagons layer by layer. For each new row there are as many degrees of freedom (lateral displacement of vertices) as there are conditions (zero area for each face).

If a honeycomb is guided by a surface  $S$  but the node axes are *not* required to be close to the normals of  $S$ , then there are many design options. When honeycombs are smooth (in an appropriate meaning of the word) then node axes will automatically be close to the normals of some developable, but that developable is present only implicitly and does not interfere with computations.



**Figure 6:** (a) Honeycomb following a near-developable surface  $S$  (which is the only kind of surface where node axes can follow the surface normals). (b) Color coded quality of a near-isometric parametrization of  $S$ , which has been used for initializing the honeycomb, by mapping a regular hexagonal tiling onto  $S$  (distortion peak .30 and average .05). (c) Spherical image of honeycomb. This result experimentally confirms that our method of initialization provides almost-isometric mappings; for details see text.



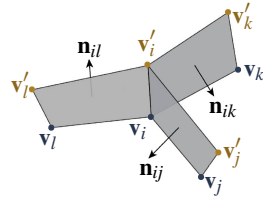
**Figure 7:** Computing a honeycomb on a complex architectural freeform skin (Yas Marina hotel, Abu Dhabi). (a) Given the base mesh, we initialize a honeycomb structure (not yet consistent) orthogonal to the reference surface and apply guided projection onto the constraint manifold. (b) This result is not satisfactorily regular, so the projection procedure has to be guided by a fairness energy. We added equality of normal vectors of opposite walls in each hexagonal cell as soft constraints (with the same weight as the regularizing term) to our algorithm. Note that the spherical image of this honeycomb exhibits Property 2.

### 3. Computational approach

Here we describe our computational setup for honeycomb structures. We discuss the two processes of (i) finding an initial hex mesh equipped with additional data which do not yet define a torsion-free support structure, and (ii) optimizing these data so they will define a honeycomb structure.

*Initialization of the base mesh.* For initialization, we implemented the method of [NPPZ12] to find a hexagonal mesh  $M = (V, E, F)$  on a given surface  $S$ , whose edges follow a given 6-RoS field of directions. To find that field, we use a direct extension of the method of [BZK09] which can easily accommodate alignment with features or a user's design strokes.

*Choice of variables.* The variables used in optimization are the vertices together with unit vectors  $\mathbf{n}_{ij}$  for each edge  $\mathbf{v}_i\mathbf{v}_j$ . In the following text we assume that  $\mathbf{n}_{ji} = -\mathbf{n}_{ij}$  but of course only one of  $\mathbf{n}_{ij}, \mathbf{n}_{ji}$  is actually used in the implementation. These vectors are initialized by letting  $\mathbf{n}_{ij} = R_{-\pi/2}(\frac{\mathbf{v}_i - \mathbf{v}_j}{\|\mathbf{v}_i - \mathbf{v}_j\|})$ , with  $R_{-\pi/2}$  as a clockwise rotation by 90 degrees about a surface normal in the midpoint of the edge  $\mathbf{v}_i\mathbf{v}_j$ . The vectors  $\mathbf{n}_{ij}$  are intended to serve as normal vectors of walls in the honeycomb, meaning that for each vertex  $\mathbf{v}_i$  there should exist a node axis  $\mathbf{v}_i\mathbf{v}'_i$  which is orthogonal to  $\mathbf{n}_{ij}$  whenever  $\mathbf{v}_i\mathbf{v}_j$  is an edge.



*Reconstructing the honeycomb structure from the chosen variables.* This is done in the following way: We must find the vertices  $\mathbf{v}'_i$  of the mesh  $M'$  paired with  $M$ , by letting  $\mathbf{v}'_i = \mathbf{v}_i + \lambda_i \mathbf{v}_i^\sigma$ , where  $\lambda_i$  is the prescribed thickness of the honeycomb and  $\mathbf{v}_i^\sigma = \frac{1}{\|\mathbf{v}'_i - \mathbf{v}_i\|}(\mathbf{v}'_i - \mathbf{v}_i)$  is a unit vector indicating the node axis through  $\mathbf{v}_i$  which must be orthogonal to all  $\mathbf{n}_{ij}$ 's. It is found by principal component analysis, as an eigenvector of the 3 by 3 matrix  $\sum_j \mathbf{n}_{ij} \mathbf{n}_{ij}^T$ .

*Constraints.* We impose the following constraints on our

variables. Firstly, every normal vector is normalized, so  $\mathbf{n}_{ij}^T \mathbf{n}_{ij} = 1$ . Secondly we have the consistency condition

$$\mathbf{n}_{ij}^T (\mathbf{v}_i - \mathbf{v}_j) = 0 \quad \text{whenever } \mathbf{v}_i\mathbf{v}_j \in E.$$

The intersection angle of walls associated with edges  $\mathbf{v}_i\mathbf{v}_j$  and  $\mathbf{v}_i\mathbf{v}_k$  could be expressed by  $\mathbf{n}_{ij}^T \mathbf{n}_{ik} = \cos 120^\circ = -1/2$ . For a valence 3 vertex  $\mathbf{v}_i$  with neighbours  $\mathbf{v}_j, \mathbf{v}_k, \mathbf{v}_l$  however, the simple condition  $\mathbf{n}_{ij} + \mathbf{n}_{ik} + \mathbf{n}_{il} = \mathbf{0}$  states that these three normal vectors form an equilateral triangle. This expresses both the angle condition and the existence of a common node axis. Further constraints are vertices confined to the design surface  $S$  or to a boundary curve. Since the geometry of the honeycomb permits moving vertices along the node axes, proximity to  $S$  can be considered a soft constraint.

*Solution.* Each constraint involves only a few variables and (apart from interpolation constraints) is either linear or quadratic. We therefore use the method of [TSG\*14] to solve the constraint equations iteratively. In every round of iteration a Newton linearization turns the constraints into a linear system of the form  $HX = R$ , with  $X$  as the increment in the vector of variables. To allow for redundant constraints and the general under-determinedness of the system, we do not solve it directly, but we regularize and instead solve  $\|HX - R\|^2 + \epsilon \|X\|^2 \rightarrow \min$ , with  $\epsilon \ll 1$ . Note that our solution procedure is essentially the same as minimizing a least-squares sum of constraints with a Gauss-Newton method.

Interpolation constraints have the form " $\mathbf{v} \in \Phi$ " where  $\Phi$  is a curve (or surface). They are linearized in each round of iteration by replacing  $\Phi$  with the tangent (or tangent plane) in the point  $\mathbf{v}^* \in \Phi$  which is closest to  $\mathbf{v}$ .

*Fairness* is not used in the basic version of our algorithm. It can be incorporated either as a set of soft constraints (with small weights) or by adding a fairness energy to the regularizing term, which amounts to the same thing (see Fig. 7).

**Relevance of geometric properties.** Section 2 discussed in detail geometric properties of honeycomb structures. Their most important implications on our algorithm (this section) and on applications (next section) are the following:

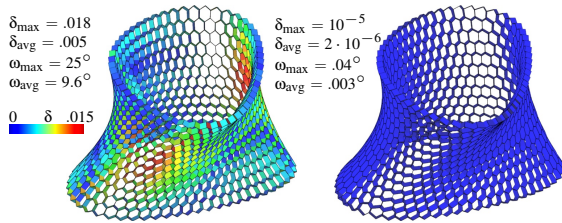
1. We cannot expect that the walls of honeycomb cells can be made orthogonal to a reference surface – this is possible only if that surface is developable (cf. Property 2 and Property 3).
2. We cannot expect the node axes (or the cell’s normal vectors) to be smooth in the usual discrete sense of finite differences, since the zero area property of Proposition 1 implies high frequency oscillations.

In particular we conclude that orthogonality of a honeycomb to a surface cannot be a hard constraint, and that fairness energies must be composed appropriately.

The next section discusses applications and extensions of the concept of honeycomb, and also mentions where it is necessary to modify the initialization or to add additional variables and constraints to optimization.

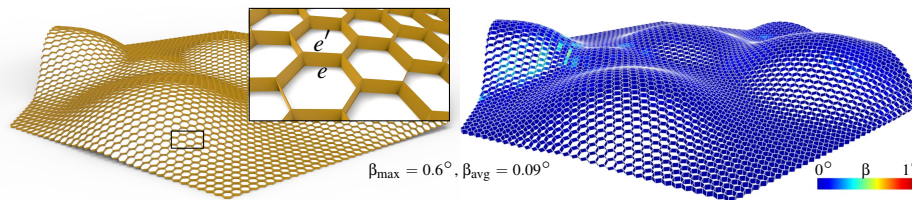
#### 4. Results

The previous sections already contained examples of honeycombs, see Figures 4, 5, 6 and 7. Property 2 is validated by the spherical images of honeycombs shown there. Further, Figure 8 gives an impression on how well the constraint equations are satisfied before and after optimization.

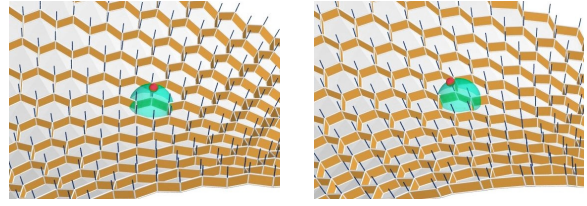


**Figure 8:** We illustrate to which extent the constraints are fulfilled, for the honeycomb of Figure 5, after initialization and before optimization (left) and after optimization (right). The planarity of wall quadrilaterals is indicated by color coding the planarity measure  $\delta$ , which is defined as the distance of diagonals of a quad divided by average edge length. “ $\omega$ ” is the deviation from the desired intersection angle of  $120^\circ$ .

**Honeycombs following developable surfaces.** Figure 6 shows a honeycomb whose guiding surface  $S$  is nearly developable and where initialization according to §3 yields an almost-isometric parametrization of  $S$ . The honeycomb



**Figure 9:** Honeycomb defined by parallel meshes  $M$  and  $M'$ . The angle  $\beta$  between corresponding edges  $e, e'$  illustrates deviation from parallelity.



**Figure 10:** An example of user guided design: The user edits the honeycomb of Figure 5 by moving a node axis. This change is also applied to neighbouring node axes, multiplied with a dampening function. Re-optimization modifies the honeycomb such that walls intersect near the prescribed node axis positions (see accompanying VIDEO).

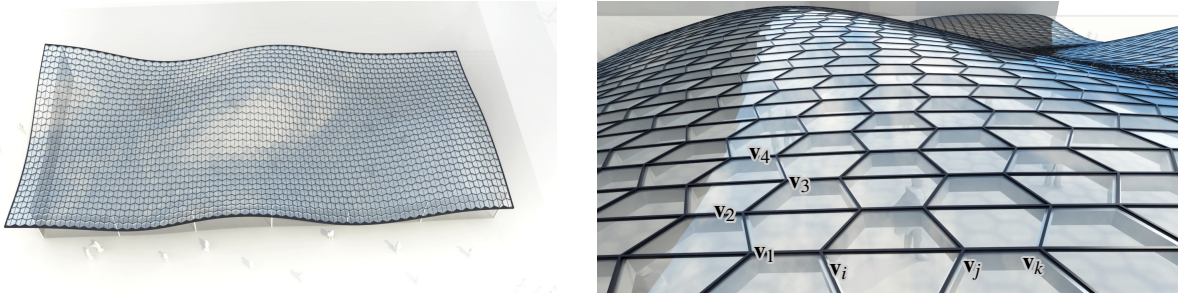
is initialized by mapping a regular hexagonal grid onto  $S$ . For the sake of experiment, we tried to make the parametrization more isometric, following [LXZ\*08] to iteratively modify the 6-RoSy field our method is based on. We found that this improvement has almost no visible effect and that the original method yields a parametrization which is isometric enough anyway.

**User interaction.** Solving the constraint equations necessary for computing a honeycomb is fast enough to enable user interaction. There are three kinds of deformation resp. modeling we emphasize: (i) changing the base mesh and re-computing the honeycomb for the modified mesh; (ii) repositioning vertices on the honeycomb’s node axes, so that the wall planes remain unchanged, and (iii) editing the axes with only minimal changes to the base mesh. All three are shown in the accompanying VIDEO. As to computation, editing mode (i) corresponds to setting target values for the vertex coordinates  $\mathbf{v}_i$  and re-running optimization. For (ii), no optimization has to be performed; this deformation is conceptually similar to editing a triangle mesh. For (iii), see Figure 10, the user’s wish is translated to new target values  $\mathbf{a}_i$  for the node axis vectors  $\frac{1}{\|\mathbf{v}'_i - \mathbf{v}_i\|} (\mathbf{v}'_i - \mathbf{v}_i)$ . They are incorporated in our optimization by adding constraints

$$\mathbf{n}_{ij}^T \mathbf{a}_i = 0, \text{ whenever } \mathbf{v}_i, \mathbf{v}_j \in E.$$

These equations express the wish that the wall planes intersect in the updated node axes. In the optimization these constraints are given a lower weight than those expressing consistency and angles.

**Honeycombs defined by parallel meshes.** We are interested in the question if there are honeycombs defined by



**Figure 11:** Zigzag quadrangulation: We show two views of a polyhedral quad mesh equipped with a honeycomb support structure, approximating the design surface of the Cour Visconti roof in the Louvre, Paris (see also Figure 1). The right hand image illustrates the “zigzag” mesh polylines (with consecutive vertices  $\mathbf{v}_1, \dots, \mathbf{v}_4$ ) and “straight” mesh polylines (with consecutive vertices  $\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k$ ). Fairness of these polylines is a topic of optimization.

parallel meshes  $M, M'$ , meaning that corresponding edges of  $M, M'$  are parallel (note that parallelity of this kind is not the same as parallelity of meshes as studied by [PLW\*07], since  $M, M'$  do not have planar faces). It is not difficult to add this constraint to our optimization. Introducing vertices  $\mathbf{v}'_i$  as additional variables, we add the appropriate consistency condition  $\mathbf{n}_{ij}^T (\mathbf{v}'_i - \mathbf{v}'_j) = 0$  for each edge  $\mathbf{v}_i \mathbf{v}_j$ , as well as the parallelity constraint  $(\mathbf{v}_i - \mathbf{v}_j) \times (\mathbf{v}'_i - \mathbf{v}'_j) = \mathbf{o}$ . In order to verify that optimization can succeed, we investigate the degrees of freedom available when, given a honeycomb, we try to position the vertices of both  $M, M'$  on their respective node axes such that edges become parallel.

Consider an  $n$ -gon cell, and assume that vertices  $\mathbf{v}_1, \dots, \mathbf{v}_{n-1}$  and  $\mathbf{v}'_1, \dots, \mathbf{v}'_{n-1}$  are chosen already. In order to construct the remaining vertices  $\mathbf{v}_n, \mathbf{v}'_n$  we observe that parallelity of corresponding edges means that the planes  $\alpha, \alpha'$  spanned by  $\mathbf{v}_1, \mathbf{v}_{n-1}, \mathbf{v}_n$  and  $\mathbf{v}'_1, \mathbf{v}'_{n-1}, \mathbf{v}'_n$ , resp., are parallel. Those planes are determined by the data already available:  $\alpha$  passes through  $\mathbf{v}_1, \mathbf{v}_{n-1}$  and is parallel to  $\mathbf{v}'_{n-1} - \mathbf{v}'_1$ , and analogous for  $\alpha'$ . We therefore find  $\mathbf{v}_n$ , resp.  $\mathbf{v}'_n$  by intersecting the given node axis with  $\alpha$ , resp.  $\alpha'$ . This shows that in a honeycomb of regular combinatorics we can find parallel meshes  $M, M'$ , simply by choosing vertices on a zigzag sequence of walls and working our way outward from there, with 1 d.o.f. per added face.

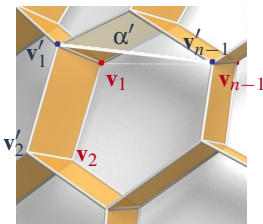


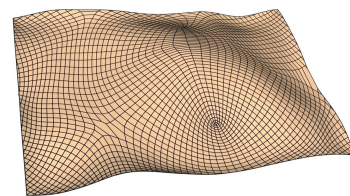
Figure 9 shows a result of this kind of optimization. The additional variables  $\mathbf{v}'_i$  have been initialized by moving the vertex  $\mathbf{v}_i$  a constant distance along the normal of the reference surface  $S$ .

**Constant honeycomb depth.** A honeycomb of constant depth is very interesting for architectural geometry, because in this case, we can let beams of constant width follow the edges of  $M, M'$ . It is not difficult to add constant

depth as a soft constraint, e.g. by adding quadratic equations  $\|\mathbf{v}_i - \mathbf{v}'_i\|^2 = c$  with a low weight. This turned out to be unnecessary for the example of Figure 9.

**Quadrangulation combined with torsion-free support structures.** For applications in freeform architecture the ability to cover a given surface by planar elements is particularly relevant. Since a honeycomb structure is usually not bounded by a hex mesh with planar faces, we demonstrate how to cover a honeycomb by a pattern of planar quadrilaterals. The resulting mesh is polyhedral, and is equipped with a torsion-free support structure (which even has the feature of repetition in local node geometry).

Our approach is to split each hexagonal face of the base mesh  $M$  into two quads which are required to be planar; thus capping each cell of the honeycomb with two planar quads. See Figures 1, 11 for the combinatorics of this splitting and a result.



**Figure 12:** It may be difficult to approximate freeform architectural designs (here: Cour Visconti, Louvre) by a quad mesh with planar faces whose edges discretize a smooth network of curves. This is because such meshes have to follow conjugate curve networks, and the number of singularities of these networks is determined by the curvature of the surface.

A major difference to previous approaches to planar-quad meshing is that we do not require smoothness of the edge polygons. In this way we no longer have the strong restrictions posed on quadrangulation which originate in an analogy between meshes and curve networks, cf. [ZSW10], e.g.

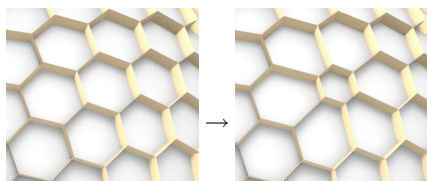
for the surface used in our example, a “smooth” quadrangulation with planar faces would have to look essentially like the one in Figure 12.

In computations we use as additional variables the unit normal vectors of those quads which are required to be planar. The vertices of these quads are already present as variables. Constraints (normalization and consistency) are analogous to the planar “wall” quads.

To capture the visual impression of smoothness (regularity) of both straight and zigzag polylines in the hex mesh  $M$ , we add soft constraints with small weight. Referring to Figure 11, a zigzag polyline in a hex mesh is considered regular, if we approximately have  $(\mathbf{v}_1 - \mathbf{v}_2) - (\mathbf{v}_3 - \mathbf{v}_4) = \mathbf{0}$ , for all choices of consecutive vertices  $\mathbf{v}_1, \dots, \mathbf{v}_4$ . For the polylines transverse to the zigzag ones we merely want to punish lateral deviation from the appearance of smoothness, meaning that we want  $\mathbf{n}_{ij}^T(\mathbf{v}_j - \mathbf{v}_k) = 0$  to hold for each choice of 3 consecutive vertices  $\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k$ . Both kinds of equations are added as soft constraints to our optimization procedure.

*Remark:* It is not difficult to add the condition of a minimum edge length  $l_{\min}$  to our optimization. This can be done by adding quadratic constraints  $\|\mathbf{v}_i - \mathbf{v}_j\|^2 = l_{\min}^2 + (\text{dummy})^2$  which use dummy variables, cf. [TSG\*14].

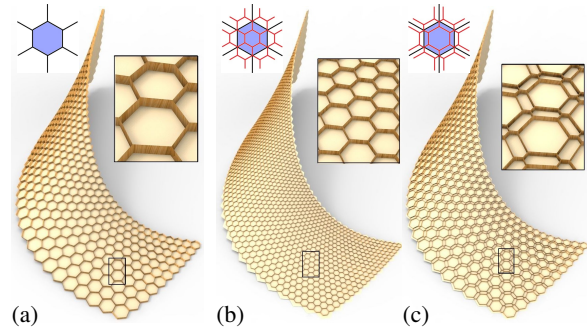
**Local edits and subdivision.** The property of two planes forming an angle of  $120^\circ$  is not destroyed by parallel translation. Likewise the intersection line of such planes may move if such a transformation is applied, but it keeps its direction. Our aim is to exploit this fact and modify honeycomb structures by parallel translating the walls. If we wish to keep the property that *three* planes are incident with a node axis, then the translations involved are not independent: If the wall planes associated with edges  $\mathbf{v}_i\mathbf{v}_j$ , resp.  $\mathbf{v}_i\mathbf{v}_k$ , resp.  $\mathbf{v}_i\mathbf{v}_l$  undergo translation by the vector  $\lambda_{ij}\mathbf{n}_{ij}$ , resp.  $\lambda_{ik}\mathbf{n}_{ik}$ , resp.  $\lambda_{il}\mathbf{n}_{il}$ , then a common intersection exists if and only if  $\lambda_{ij} + \lambda_{ik} + \lambda_{il} = 0$  (because the three normal vectors form an equilateral triangle).



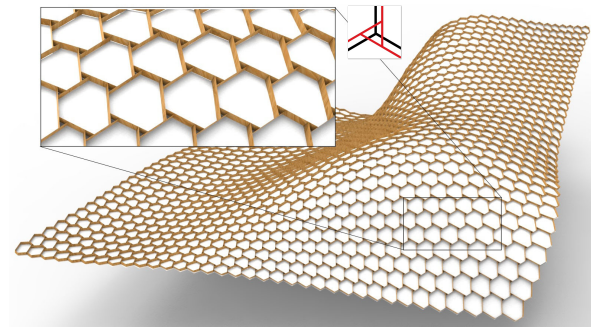
**Figure 13:** Resizing a honeycomb cell by moving walls inwards by the same amount.

One example of such a move is the resizing of a single honeycomb cell (Figure 13): here all walls of the cell move by the same amount, while the “radial” walls stay where they are (in the vertex-centered notation used above, we have e.g.  $\lambda_{ij} = 0$ ,  $\lambda_{ik} = -\lambda_{il}$ ).

By resizing *all* cells, we create new hexagonal cells where



**Figure 14:** Subdivision of honeycombs by graphical stencils. There are several ways we can parallel translate walls of a honeycomb so that intersection properties are maintained, and which are useful for refinement and the definition of other derived structures. These modifications are encoded in a symbolic way by a sketch of a cell or node which shows the old and new positions of walls. (a), (b) The symbol  $\text{hexagon} \rightarrow \text{subdivided hexagon}$  signifies subdivision, generating a regular honeycomb with 1 new cell per old cell, and 1 new cell per old wall, with all new cells roughly the same size. (c) The analogous symbol  $\text{hexagon} \rightarrow \text{subdivided hexagon}$  signifies subdivision which is combinatorially the same but geometrically different, where the newly created cells are of different sizes.



**Figure 15:** Using the language of Figure 14, the symbol  $\text{node} \rightarrow \text{triangle}$  signifies the creation of a reciprocal structure, replacing each node by a triangular cell.

the original walls had been: this is the subdivision procedure shown by Figure 14. Shrinking each cell of the honeycomb in Figure 14a to half its size (or another size) produces the subdivided honeycomb of Figure 14b (or Figure 14c). Its spherical image coincides with the original one, but it has 4 times as many cells.

*Remark:* For a general torsion-free support structure, where planes intersect at different angles, local edits (and in consequence, subdivision) are not so easy. Each node will have its own condition on the three translations which affect that node, and those conditions will generally not be consistent when cycling around a cell.



Fig.	$ V $	$ E $	$w_{\text{angle}}$	$w_{\text{cons}}^{\text{wall}}$	$w_{\text{norm}}$	$w_{\text{fair}}^{\text{zig}}$	$w_{\text{fair}}^{\text{str}}$	$w_{\parallel}$	$w_{\text{cons}}^{\text{faces}}$	$w_{\text{axes}}$	# It	$\omega_{\text{max}}$	$\omega_{\text{avg}}$	$\delta_{\text{max}}^{\text{wall}}$	$\delta_{\text{avg}}^{\text{wall}}$	$\delta_{\text{max}}^{\text{face}}$	$\delta_{\text{avg}}^{\text{face}}$	$T$
5,8	1512	2214	1	1	10						5	.037	.003	$1.2 \cdot 10^{-5}$	$2.2 \cdot 10^{-6}$			0.6
7	8497	12434	1	1	10						5	.91	.04	$7.0 \cdot 10^{-4}$	$7.3 \cdot 10^{-6}$			4.1
9	6144	9120	1	1	10			1			5	.073	.012	$1.3 \cdot 10^{-5}$	$6.6 \cdot 10^{-5}$			4
11,1	3501	5165	1	1	10	.01	.01		.1		5	.051	.008	$1.5 \cdot 10^{-5}$	$2.5 \cdot 10^{-6}$	.007	.003	3
14a	958	1386	1	1	10						10	.30	.028	$1.3 \cdot 10^{-4}$	$1.4 \cdot 10^{-5}$			1.1
15	3198	4718	1	1	10						5	.37	.016	$2.1 \cdot 10^{-4}$	$3.4 \cdot 10^{-6}$			2
18	3293	4827	1	1	10					.1	5	.67	.16	$1.8 \cdot 10^{-3}$	$8 \cdot 10^{-4}$			2

**Figure 16:** Statistics and parameters. We show the weights given to constraint equations ( $w_{\text{angle}}$  for the angle condition,  $w_{\text{cons}}^{\text{wall}}$ ,  $w_{\text{cons}}^{\text{faces}}$  for consistency of wall quads and other quads,  $w_{\text{norm}}$  for normalization,  $w_{\text{fair}}^{\text{zig}}$ ,  $w_{\text{fair}}^{\text{str}}$  for fairness,  $w_{\parallel}$  for parallelity of meshes,  $w_{\text{axes}}$  for prescribed node axes), as well as the results' deviation  $\omega$  from the desired intersection angle  $120^\circ$  and the planarity measure  $\delta$  (cf. Figure 8). We further give the number of iterations needed for computation and the time in seconds.

A procedure destroying node axes is shown by Figure 15: the walls of hexagonal cells are moved outward and inward in an alternating way, defining a *reciprocal structure*, cf. [SFG\*13]. This structure is still formally a honeycomb, if we allow T-junctions.

**Shading Systems.** Wang et al. [WJB\*13] have proposed shading systems as an application of discrete line congruences. It is in fact not difficult to optimize honeycomb structures so that they serve the same purpose, such that the walls of cells block light effectively while the depth of the honeycomb remains small. We define a vector  $\mathbf{l}$  which corresponds to the direction of light, e.g. at 1 p.m. in summer. Since the angles between walls are always  $120^\circ$  it would not make sense to optimize the honeycomb such that walls are orthogonal to  $\mathbf{l}$ . Instead we look for any field of unit vectors  $\mathbf{a}_i$  attached to the individual vertices  $\mathbf{v}_i$  which are orthogonal to  $\mathbf{l}$  and which serve as prescribed directions of node axes. We then optimize the honeycomb with constraints  $\mathbf{a}_i^T \mathbf{n}_{ij} = 0$  (for all edges  $\mathbf{v}_i \mathbf{v}_j$ ). In the optimization these constraints are given a lower weight than those expressing consistency and angles. A result can be seen in Figure 18.

## 5. Discussion

The success of the regularized Newton method used here has already been demonstrated by [TSG\*14]. Referring to that paper, we mention that it works well for constraints which are linear or quadratic, and which do not involve many variables. Similar to [TSG\*14] we obtain high accuracy. Figure 16 gives details on optimization and geometric properties of the results. Timings refer to an Intel Xeon CPU with 2.67GHz (we also mention that as a sparse linear solver we used the TAUCS library).

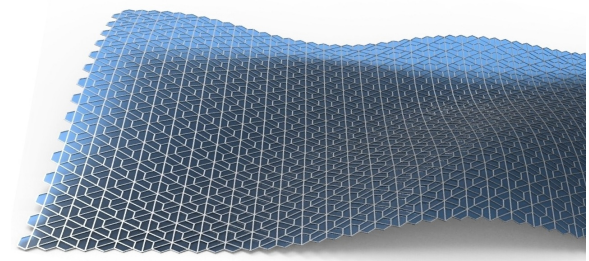
**Limitations.** The main geometric limitation of honeycombs has been discussed in §2: it is the property that  $120^\circ$  angles between walls imply that the honeycomb's node axes do not in general follow the surface normals of a given reference shape. If we do not insist on this condition (which we could

do only for developable surfaces), honeycombs have rather many degrees of freedom, and we found no obstructions in our numerical experiments.

From the viewpoint of statics, structures based on hexagonal meshes are of course more difficult than structures based on triangle meshes, for the simple reason that there are fewer load-bearing edges. In practice, auxiliary force-transmitting elements like cables are typically used. We do not consider this aspect there.

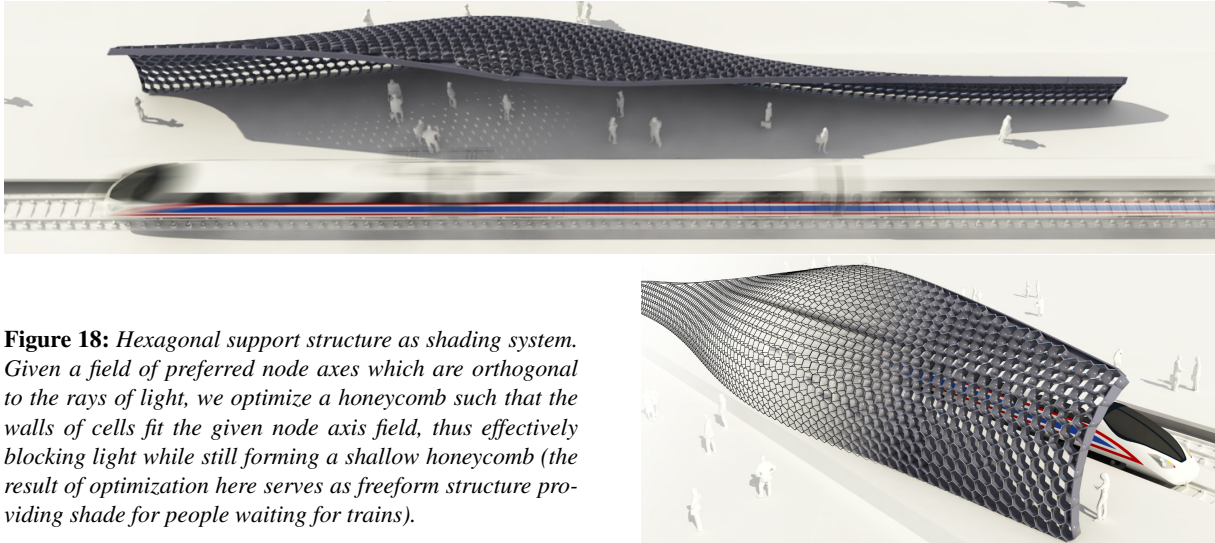
## 6. Conclusion

We have presented geometry and computation of honeycomb structures and their applicability to various tasks in freeform architectural design. Notably honeycombs provide a torsion-free support structure with identical nodes, but can be used even for quadrangulation.



**Figure 17:** A pattern of planar quadrilaterals derived from a honeycomb structure in much the same manner as Figure 11, but using a different set of diagonals through cells.

**Future Research.** As topics of future research we would like to propose the investigation of polyhedral meshes whose edges and faces are arranged in more general patterns, see Figure 17. This requires to deviate from traditional notions of smoothness, and to use this deviation as a design element. Already in this paper we have seen that one gains additional degrees of freedom in this way.



**Figure 18:** Hexagonal support structure as shading system. Given a field of preferred node axes which are orthogonal to the rays of light, we optimize a honeycomb such that the walls of cells fit the given node axis field, thus effectively blocking light while still forming a shallow honeycomb (the result of optimization here serves as freeform structure providing shade for people waiting for trains).

### Acknowledgments

We are grateful to Marko Tomičić for renderings. This research was supported by KAUST base funding, the TU Graz “NAWI” project, by the DFG-Collaborative Research Center, TRR 109 *Discretization in Geometry and Dynamics*, through grants I 705 N-26 and I 706-N26 of the Austrian Science Fund (FWF), by FWF project P 23735-N13 and by the European Community’s 7th Framework Programme under grant agreement 286426 (GEMS).

### References

- [BPK\*11] BO P., POTTMANN H., KILIAN M., WANG W., WALLNER J.: Circular arc structures. *ACM Trans. Graph.* 30 (2011), #101,1–11. Proc. SIGGRAPH. doi:10.1145/1964921.1964996.
- [BZK09] BOMMES D., ZIMMER H., KOBELT L.: Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3 (2009), #77,1–10. Proc. SIGGRAPH. doi:10.1145/1531326.1531383.
- [FLHCO10] FU C.-W., LAI C.-F., HE Y., COHEN-OR D.: K-set tilable surfaces. *ACM Trans. Graph.* 29 (2010), #44,1–6. Proc. SIGGRAPH. doi:10.1145/1833349.1778781.
- [LSZ\*14] LU L., SHARF A., ZHAO H., WEI Y., FAN Q., CHEN X., SAVOYE Y., TU C., COHEN-OR D., CHEN B.: Build-to-last: Strength to weight 3D printed objects. *ACM Trans. Graph.* 33, 4 (2014). Proc. SIGGRAPH. doi:10.1145/2601097.2601168.
- [LXZ\*08] LIU L., XU Y., ZHANG L., GOTSMAN C., GORTLER S. J.: A Local/Global Approach to Mesh Parameterization. *Computer Graphics Forum* 27, 5 (2008), 1495–1504. doi:10.1111/j.1467-8659.2008.01290.x.
- [NPPZ12] NIESER M., PALACIOS J., POLTHIER K., ZHANG E.: Hexagonal global parameterization of arbitrary surfaces. *IEEE Trans. Vis. Comp. Graphics* 18, 6 (2012), 865–878. doi:10.1109/TVCG.2011.118.
- [PJH\*14] POTTMANN H., JIANG C., HÖBINGER M., WANG J., BOMPAS P., WALLNER J.: Cell packing structures. *Computer-Aided Design* (2014). doi:10.1016/j.cad.2014.02.009.
- [PLW\*07] POTTMANN H., LIU Y., WALLNER J., BOBENKO A., WANG W.: Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graph.* 26 (2007), #65,1–11. Proc. SIGGRAPH. doi:10.1145/1275808.1276458.
- [SEKC13] SCHIFTNER A., EIGENSATZ M., KILIAN M., CHINZI G.: Large scale double curved glass facades made feasible – the Arena Corinthians west facade. In *Glass Performance Days Finland (Conference Proceedings)*. 2013, pp. 494 – 498. [www.glassfiles.com].
- [SFG\*13] SONG P., FU C.-W., GOSWAMI P., ZHENG J., MITRA N., COHEN-OR D.: Reciprocal frame structures made easy. *ACM Trans. Graph.* 32 (2013), #94,1–10. Proc. SIGGRAPH. doi:10.1145/2461912.2461915.
- [SHWP09] SCHIFTNER A., HÖBINGER M., WALLNER J., POTTMANN H.: Packing circles and spheres on surfaces. *ACM Trans. Graph.* 28 (2009), #139,1–8. Proc. SIGGRAPH Asia. doi:10.1145/1618452.1618485.
- [SS10] SINGH M., SCHAEFER S.: Triangle surfaces with discrete equivalence classes. *ACM Trans. Graph.* 29 (2010), #46,1–7. Proc. SIGGRAPH. doi:10.1145/1778765.1778783.
- [TSG\*14] TANG C., SUN X., GOMES A., WALLNER J., POTTMANN H.: Form-finding with polyhedral meshes made simple. *ACM Trans. Graphics* 33, 4 (2014). Proc. SIGGRAPH. doi:10.1145/2601097.2601213.
- [WJB\*13] WANG J., JIANG C., BOMPAS P., WALLNER J., POTTMANN H.: Discrete line congruences for shading and lighting. *Computer Graphics Forum* 32 (2013), 53–62. Proc. Symp. Geom. Processing. doi:10.1111/cgf.12172.
- [ZCBK12] ZIMMER H., CAMPEN M., BOMMES D., KOBELT L.: Rationalization of triangle-based point-folding structures. *Computer Graphics Forum* 31 (2012), 611–620. Proc. Eurographics. doi:10.1111/j.1467-8659.2012.03040.x.
- [ZSW10] ZADRAVEC M., SCHIFTNER A., WALLNER J.: Designing quad-dominant meshes with planar faces. *Computer Graphics Forum* 29 (2010), 1671–1679. Proc. Symp. Geom. Processing. doi:10.1111/j.1467-8659.2010.01776.x.