ELSEVIER

Contents lists available at ScienceDirect

Computer-Aided Design



journal homepage: www.elsevier.com/locate/cad

Disjointed force polyhedra*



Juney Lee *, Tom Van Mele, Philippe Block

ETH Zurich, Institute of Technology in Architecture, Block Research Group, Stefano-Franscini-Platz 1, HIB E 45, CH-8093 Zurich, Switzerland

ARTICLE INFO

Article history: Received 3 November 2017 Accepted 10 February 2018

Keywords:

Three-dimensional graphic statics Polyhedral reciprocal diagrams Extended Gaussian image Polyhedral reconstruction Spatial equilibrium Constrained form finding

ABSTRACT

This paper presents a new computational framework for 3D graphic statics based on the concept of disjointed force polyhedra. At the core of this framework are the Extended Gaussian Image and areapursuit algorithms, which allow more precise control of the face areas of force polyhedra, and consequently of the magnitudes and distributions of the forces within the structure. The explicit control of the polyhedral face areas enables designers to implement more quantitative, force-driven constraints and it expands the range of 3D graphic statics applications beyond just shape explorations. The significance and potential of this new computational approach to 3D graphic statics is demonstrated through numerous examples, which illustrate how the disjointed force polyhedra enable force-driven design explorations of new structural typologies that were simply not realisable with previous implementations of 3D graphic statics.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Recent extensions of *graphic statics* to three dimensions have shown how the static equilibrium of spatial systems of forces can be represented by a *polyhedral force diagram*, or closed *force polyhedra*. In 3D graphic statics, the areas and normals of the faces of the force polyhedra represent the magnitudes and directions of the corresponding forces in the system [1,2]. With explicit, bidirectional control of both the geometry of the structural form and its force equilibrium, polyhedral reciprocal diagrams can be used for a wide range of design applications. Most notable implementations include equilibrium analysis, design and form finding of complex spatial geometries through various subdivision schemes [3] and additive transformations [4] of force polyhedra.

1.1. Problem statement

One of the most powerful benefits of computational graphic statics is that it enables dynamic interaction between the *reciprocal form and force diagrams* with real-time, visual feedback [5]. In 2D graphic statics, the length of an edge in the force diagram represents the magnitude of the axial force in the corresponding member of the structure. Therefore, changing the geometry of the force diagram has a direct influence on the magnitude of the internal forces. The vertices of 2D force diagrams can be geometrically constrained to explore force-constrained structural forms,

such as constant-force trusses [6]. However, in 3D graphic statics using polyhedral reciprocal diagrams, the influence of changing the geometry of the force polyhedra on the force magnitudes is not as direct or intuitive. For example, modifying the locations of the vertices of the force polyhedra changes the areas of the polyhedral faces, but there is no clear visual relationship between the operation and the resulting face areas of the polyhedra. The lack of means to explicitly and precisely control the face areas of the force polyhedra (or equivalently, the magnitudes of the axial forces within the corresponding members) limits 3D graphic statics applications to abstract shape explorations. Subsequently, it is difficult to incorporate quantitative, force-driven considerations during the design process, which is necessary for addressing more realistic structural design problems and boundary condition constraints.

Additionally, current computational implementations of 3D graphic statics require that every constituent polyhedron or a polyhedral *cell* of the force polyhedra, is constrained to have the same contact face geometries and areas with each of its neighbouring cells. This geometric constraint of the contact faces intrinsically enforces the area constraint, and therefore it is convenient for constructing consolidated force polyhedra and applying global geometric subdivisions and transformations. However, design explorations using consolidated force polyhedra strictly limit the corresponding form diagrams to also be geometrically polyhedral (i.e. subdivided tree structures, faceted domes, polyhedral mesh or surface structures, crystalline aggregations, etc.). In addition, the contact face geometry constraint makes it impossible to construct consolidated force polyhedra for equilibrated structures that are non-polyhedral (i.e. structures with overlapping members, nonplanar faces, etc.). The range of structures that can currently be

This paper has been recommended for acceptance by Helmut Pottmann.
 Corresponding author.

E-mail addresses: juney.lee@arch.ethz.ch (J. Lee), van.mele@arch.ethz.ch (T.V. Mele), block@arch.ethz.ch (P. Block).

explored with polyhedral reciprocal diagrams is therefore limited to polyhedral geometries only.

1.2. Objectives

In order to expand the range of design applications using polyhedral reciprocal diagrams beyond abstract shape explorations, a new computational framework that enables the control of polyhedral face areas is needed. In current implementations of 3D graphic statics, polyhedral force diagrams are computationally constructed as volumetric meshes with matching contact faces and shared vertex coordinates between any pair of adjacent cells. In the proposed approach, any pair of adjacent cells are constrained to have the same contact face orientations and areas, but each with its own vertex locations and topology.

The new computational framework for 3D graphic statics presented in this paper will provide the necessary datastructure, functionalities and algorithms to enable designers to interactively and intuitively control the face areas of the force polyhedra. The ability to explicitly control the polyhedral face areas allows integration of quantitative, force-driven constraints and exploration of new typologies that are not realisable with previous applications of 3D graphic statics. This new framework is developed and presented with the ultimate goal of maintaining and improving the inherent and most important benefits of computational graphic statics: legible visualisation of force equilibrium, intuitive designer interactivity, and provision of new structural design insights.

1.3. Contributions and outline

The outline of the paper is as follows.

In Section 2, we introduce the concept of disjointed force polyhedra and present the relevant theoretical background. Various polyhedral reconstruction techniques from different disciplines are summarised and discussed, with emphasis on methods based on polyhedral face normal and areas.

In Section 3, we provide an overview of the computational implementation of the presented approach to 3D graphic statics. The algorithms for reconstructing or modifying a polyhedron with specified face normals and areas are described in detail with illustrations and condensed code snippets. A new datastructure is also presented, which synthesises the above-mentioned algorithms and functionalities to seamlessly integrate the presented approach in an interactive design environment.

In Section 4, the design potential of the presented framework is demonstrated through several examples to highlight the new structural typologies that can be explored, and force-driven design applications using disjointed force polyhedra.

We conclude the paper with a discussion on the practical potential of the presented approach, and its relevance to other spatial form-finding methods such as the *Force Density Method* (FDM) [7] and *Thrust Network Analysis* (TNA) [8].

2. Theoretical background

This section describes the underlying principles of the concept of disjointed force polyhedra.

First, we briefly review the basic properties of polyhedral reciprocal diagrams and establish the nomenclature that will be used throughout the paper. Next, we graphically show how any pair of adjacent cells of a force polyhedra can be interfaced with a *collapsed cell* corresponding to a fictitious node in the structure, thereby allowing the two initial cells to have contact faces that are different in geometry but have the same areas. We then give a brief overview of previous work on various polyhedral reconstruction methods, and relevant techniques that allow computation



Fig. 1. (a) Γ of node **v**_{*i*} in equilibrium; (b) the sum of all the internal force vectors at the node (for a more detailed explanation of this proof, we refer the readers to [2, p. 120]); and (c) Γ^{\perp} and the corresponding cell, c_i .

of polyhedral geometry from face normals and areas. Polyhedral reconstruction techniques previously presented within the context of 3D graphic statics are also reviewed. We conclude the section by summarising the limitations of the current state of the art, and identifying key areas for improvements.

2.1. Polyhedral reciprocal diagrams

In 3D graphic statics, the equilibrium of the external forces or the *i* th node \mathbf{v}_i of a structure (Fig. 1a) is represented by a closed polyhedron or a polyhedral cell, c_i (Fig. 1c). For each cell c_i , the normal $\hat{\mathbf{n}}_{i,j}$ and the area $A_{f_{i,j}}$ of the *j* th face $f_{i,j}$ represent the direction and magnitude of the axial force $\mathbf{f}_{i,j}$ in the corresponding member or the *edge* $e_{i,j}$ of the polyhedral form diagram, respectively. The interpretation of $\mathbf{f}_{i,j}$ at node \mathbf{v}_i as either compression or tension can be made by comparing the polyhedral face normal $\hat{\mathbf{n}}_{i,j}$ and the orientation of the corresponding edge in the form diagram [9,4].

In this paper, Γ will refer to a polyhedral form diagram, which represents a pin-jointed spatial truss in equilibrium that is polyhedral in its geometry. Γ^{\perp} will refer to a polyhedral force diagram or a consolidated force polyhedra, which is both dual and reciprocal to Γ .

2.2. Disjointed force polyhedra

Previously presented design explorations using 3D graphic statics have been based on aggregations, subdivisions or transformations of polyhedral force diagrams where all pairs of adjacent cells have matching contact faces (Fig. 2a), and therefore can be assembled into a consolidated force polyhedra, Γ^{\perp} . The concept of neighbouring cells with dissimilar or mismatching contact faces was first introduced by McRobie [10] (Fig. 2b). In more recent papers, McRobie [11,12] showed that the equilibrium of two neighbouring cells with mismatching contact face geometries can be explained using "face cushions" or collapsed cells. The "face cushions" represent fictitious nodes in Γ with a net force of zero at that node, which is important in constructing Γ^{\perp} for more complicated structures. The addition of a collapsed cell in-between two adjacent cells allows the two initial cells to become *disjointed*; the two adjacent cells need to have contact faces that are equal in area, but do not necessarily have the same geometry.

While the collapsed cells are necessary for graphically representing and describing the concept of *disjointed force polyhedra*, Ψ^{\perp} , they do not need to be visualised for practical purposes in an interactive design environment. Especially for a Ψ^{\perp} that contains a large network of cells, the collapsed cells do not need to be computed or represented, as it would only cause additional visual



Fig. 2. (a) Two adjacent nodes in equilibrium where the two corresponding cells have matching contact faces; (b) two adjacent nodes in equilibrium where the two corresponding cells are disjointed with mismatching contact faces; and (c) the addition of fictitious node \mathbf{v}_2 , and the corresponding cell c_2 linking the two mismatching contact faces; in convex and collapsed states.

clutter and consume unnecessary computational time. Rather, in order to digitally implement Ψ^{\perp} , a robust computational setup and algorithms are needed, that ensure that the contact faces of every pair of adjacent cells have the same area and parallel orientations.

Note that the corresponding form diagram Ψ of a disjointed force polyhedra Ψ^{\perp} can be in equilibrium, but not necessarily polyhedral in its geometry. Ψ and Ψ^{\perp} are also not necessarily topological duals, which means they cannot be defined as reciprocal; Ψ^{\perp} is a collection of cells that are individually reciprocal to its corresponding nodes in Ψ . The polyhedral form and force diagrams Γ and Γ^{\perp} are mutually dual and reciprocal, and are special cases of a non-polyhedral form diagram Ψ and disjointed force polyhedra Ψ^{\perp} , respectively.

2.3. Polyhedral reconstruction

Computation of polyhedral geometry, or more commonly known as *polyhedral reconstruction*, is a well-researched topic for a variety of applications in many disciplines, such as computer vision, computational geometry and combinatorics. In most applications, the objective is to reconstruct the polyhedral geometry from partial information about the polyhedra (i.e. from projected images, vertex locations, edge lengths, face geometries, face normals, face areas, dihedral angles, etc.) [13]. The polyhedral reconstruction method that is most relevant to 3D graphic statics and the objectives of this paper, is the one based on face normals and areas.

2.4. Reconstruction from face normals and areas

The theory of polyhedral reconstruction from its face normals and areas, or modifying the polyhedral geometry with target face areas, originate from Minkowski's theorem [14]. Alexandrov's interpretation of the theorem is recited below, with modified notations to stay consistent with the nomenclature used in this paper:

If $\hat{\mathbf{n}}_0, ..., \hat{\mathbf{n}}_m$ are non-coplanar unit vectors and $A_0, ..., A_m$ are positive numbers such that $\sum_{j=0}^m A_j \cdot \hat{\mathbf{n}}_j = 0$, then there exists a closed convex polyhedron whose faces have outward normals $\hat{\mathbf{n}}_j$ and areas A_j , [... with uniqueness up to translation] [15, p. 311].

Although proofs for this existence theorem can be found in numerous texts in the literature [15–17], the reconstruction procedure was never explicitly mentioned or developed in detail.

It was not until the 1980s that this problem was revisited with adequate computational tools. Ikeuchi [18] first proposed a technique by using a constrained minimisation procedure, followed by Little's [19] iterative minimisation solver using the *Extended Gaussian Image* (EGI), which is a topological representation of a surface or a polyhedral object on a unit sphere [20]. Moni [21] added another layer to the EGI-based technique by using zero-area faces to address indeterminate face adjacencies. Xu and Suk [22] introduced hierarchical EGI to reconstruct concave polyhedra. In these implementations, while robust in their theory and setup, the methods were demonstrated on only a few simple examples. The general computational complexity and hardness of this reconstruction problem was addressed in [23].

Especially with EGI-based methods, the complexity and diversity of polyhedra that can be reconstructed were not demonstrated through a large sampling of convincing examples. Furthermore, the ultimate goal of these methods is to simply demonstrate the improved efficiency over its predecessors, rather than to manipulate or interact with the resulting geometry of the polyhedra. The designers' ability to customise the method and control the computed geometry of the polyhedra is not addressed, which is crucial in an interactive design environment for architecture and structural design.

In a more numerical approach, Lachand-Robert and Oudet [24] presented a variant of a convex hull method that could reconstruct convex bodies with more than 1000 given face normals and areas. Such powerful numerical methods are necessary for large optimisation problems in various engineering applications. However, in the context of architectural and structural design using polyhedral form and force diagrams, individual cells have relatively low number of faces; a structure would typically not have nodes where more than six members come together. Instead, structures are typically a large network of simple cells. Rather than a powerful solver that can reconstruct a single cell with a large number of faces, a flexible and interactive setup that can control a large network of simple cells is needed.

2.5. Polyhedral reconstruction in 3D graphic statics

As mentioned above, previous design applications of polyhedral reciprocal diagrams have been largely based on geometric transformations of a single cell or a network of cells, then computing Γ using the *reciprocation* (\perp) *algorithm* [2,4]. The initial Γ^{\perp} was typically assumed to be a pre-existing condition (i.e. abstract solid objects modelled manually or generated parametrically in CAD software). There have been two main contributions within the context of 3D graphic statics with regards to reconstruction and generation of Γ^{\perp} .

The first method is based on translation of procedural techniques used in 2D graphic statics to 3D, using resultant forces, trial funiculars and "closing planes" to construct the *global force* polyhedron, Λ^{\perp} [25,26]. Λ^{\perp} represents the equilibrium of only the external loads and reaction forces in the form diagram, Λ . Although the step-by-step procedure is important for teaching and explaining the principles of graphic statics, repeated reconstruction and constant redrawing of trial funiculars for every node of the structure would be cumbersome and undesirable. Furthermore, the construction method is entirely geometric, which means that the face areas cannot be controlled explicitly.

The second method is based on projective geometry, which is aligned with the founding principles of graphic statics and therefore provides some invaluable insights. Using conic sections and paraboloids of revolution, Γ^{\perp} can be directly computed and drawn for spatial structures with complex topologies [27]. This method is also important in explaining the geometric relationships between n-dimensional reciprocal diagrams and their dual (n+1)dimensional Airy stress functions [28,29]. However, similarly to the procedural geometric approach, this method has not yet shown how the face areas of the polyhedra can be controlled during the reconstruction process.

3. Computational setup

This section provides an overview of the computational implementation of the proposed approach.

Three algorithms will be presented: the *Extended Gaussian Im*age (*EGI*) algorithm, the area pursuit (*AP*) algorithm, and the unified diagram (*UD*) algorithm. The EGI algorithm constructs a flexible datastructure for cells without initially knowing its final face adjacencies. Once the flexible datastructure is established, the AP algorithm re-sizes the faces of the cell towards their target areas, without having to reconstruct the datastructure at every iteration. These two algorithms together provide a method for not just reconstructing cells from prescribed face normal and areas, but also modify existing cells with new target face areas. Condensed code snippets of the algorithms with illustrations will be provided.

A datastructure for disjointed force polyhedra will be presented, which is in essence a nested network of closed meshes. While the EGI and AP algorithms operate on individual cells, this new datastructure allows implementations of these algorithms for a larger network of cells.

We conclude the section by demonstrating how the unified diagrams can be constructed for disjointed force polyhedra using the UD algorithm, thereby increasing the legibility and the potential insights and intuition the designers can gain from the interactive visualisations.

3.1. General approach

Previous work in polyhedral reconstruction have been generally based on optimisation schemes that seek to maximise the processing efficiency or the output capacity of the method itself. Therefore, these methods are not ideal for a design environment where the geometry of the force polyhedra may require constant interaction and change.

The algorithms presented are based on geometric iterative solvers which are more easily customisable, to address a large network of simple cells rather than a single cell with a large number of faces. Although the presented approach only addresses convex cells, we later show that the use of concave and complex cells can generally be avoided all together.

The presented approach is developed using the *COMPAS* library, an open-source computational framework for collaboration and research in architecture, engineering and digital fabrication [30]. Written in the Python scripting language [31] without any dependencies, the presented approach can be implemented with any desired CAD software.

3.2. EGI algorithm

EGI is a representation of surfaces or solid objects on a unit, *Gaussian sphere* using its face orientations and areas [20]. If the tails of the unitised normals of the faces are placed at the centre of the Gaussian sphere, then the heads of the unitised normals lie on the surface of the Gaussian sphere. The face areas, whether already known or input as targets, are placed as *point masses* at the heads of the corresponding normals on the Gaussian sphere (Fig. 3b). An equivalent representation is the *spike model* of the EGI, where the normal vectors are scaled by their corresponding area values (Fig. 3a).

Based on the duality principle of projective geometry, which maps faces into points, points into faces and edges to edges, the EGI is then a spherical *dual image* or topological dual of a convex polyhedron [32] (Fig. 3c). A polyhedron with *n* number of faces will have *n* number of point masses on the Gaussian sphere. Since any two faces of a polyhedron are adjacent if they share a common edge, an edge of a polyhedron can be represented on the EGI as a connection between two point masses as an *arc*. This arc is defined as the minor arc of the great circle containing any pair of points on the EGI [21]. It is possible for two points that are antipodal with each other to be connected, for example in the cases of a dihedron or a flattened polyhedron. However, such instances are not considered in this paper, and any two diametrically opposite points are assumed to be not adjacent.

However, the face adjacency information of a polyhedron or a cell is not directly recoverable from the location of the point masses and arcs alone. Consider the three cells shown in Fig. 4. These three cells have the same face orientations but different face areas. Based on the definition of an arc stated above, the face normals and areas of the three cells result in the same EGIs (middle row of Fig. 4). However, the actual EGIs of the cells (bottom row of Fig. 4) are different, as the different face area distributions result in different face adjacencies. Depending on the face area distribution, various face adjacencies occur: faces 1–4 are all adjacent with one another at a vertex; faces 1 and 3 are adjacent; or faces 2 and 4 are adjacent.

This indeterminacy of face adjacencies occurs where an arc crosses another arc on the EGI. Moni [21] defines these arc intersections as *cross-adjacencies*, where various face adjacencies could occur depending on different face area distributions. By adding a fictitious, *zero point mass* at these arc intersections, and subsequently a face with zero-area, or a *zero face*, all possible face adjacencies can be embedded and represented in a single EGI (Fig. 5a). Once all zero point masses have been added, the datastructure of the EGI as the dual spherical polyhedron is complete, and the *unit cell* can now be constructed, which is simply the topological primal of the dual spherical polyhedron (Fig. 5b). All zero faces have a target area of zero, and will eventually collapse to either an edge or a vertex as a result of the AP algorithm, which will be described in the next section.

One of the most remarkable properties of EGI is that the centre of mass of the EGI's point masses has to lie at the origin of the Gaussian sphere [20]. This means that there cannot exist a hemisphere on the EGI that does not have a point mass, which would represent an unbounded polyhedron. However, there are commonly used node elements in structural design for which the corresponding EGI may have one or multiple empty hemispheres, such as: a 2D node (Fig. 6a) where all members at that node are coplanar; an open node (Fig. 6b); or a node with members that may temporarily be unequilibrated during the form-finding process (Fig. 6c).

For these special cases, *virtual faces* are introduced to complete the geometric reconstruction of the cells. Virtual faces are not the same as zero faces, and have no target area values; the only purpose of the virtual faces is to facilitate the geometric construction of cells in these special situations, and have no corresponding member



Fig. 3. (a) Spike model of equilibrated force vectors; (b) normalised force vectors and the point masses on the Gaussian sphere; (c) EGI with adjacency arcs, or equivalently the dual spherical polyhedron; (d) spherical polyhedron, which is the topological primal of the EGI; and (e) the geometry of the polyhedron or cell.



Fig. 4. Three cells with same face orientations, but with different areas: (a) faces 1 through 4 are adjacent with one another at a single vertex; (b) faces 1 and 3 are adjacent along an edge; and (c) faces 2 and 4 are adjacent along an edge.



Fig. 5. (a) The EGI with zero point mass 5 added; and (b) the corresponding unit cell with an added zero face.

in Γ . In general, if the centre of mass is not at the origin of the Gaussian sphere, or if there are any empty hemispheres, virtual point masses are placed at the centre of the empty hemispheres.

In the case of a 2D node, only two virtual faces are required to complete the prism-like cell, with an arbitrary scale factor of ζ [4] (Fig. 6a). For an open node, at least three virtual faces are required to complete the cell (Fig. 6b). During the interactive design process, some cells could become temporarily unbounded.



Fig. 6. (a) A 2D node, its EGI and the cell with two added virtual faces; (b) an open node, its EGI and the cell with four added virtual faces; and (c) a temporarily unequilibrated node with one added virtual face, which in this case is also a zero face.

A virtual face can be added for each empty hemisphere, and only in such unbounded cases, the virtual faces will be treated as zero faces.

Using the *COMPAS* framework, a cell is represented with a Mesh class, which is an implementation of a half-edge datastructure (Fig. 9b). The EGI is topologically dual to the cell. Therefore, it can also be represented by a Mesh (Fig. 9a-1). During the construction of the EGI, geometric arcs are temporarily used to determine the points of adjacencies and cross-adjacencies (Fig. 9a-2, 3). Once all the points of cross-adjacencies are found, topological edges can be added to the datastructure. Depending on the node type and whether there are any empty hemispheres, virtual faces are appropriately added (Fig. 9a-4). With all the edges added, all the faces of the mesh can be found (Fig. 9a-5). An EGI datastructure with vertex, edge and face information can then be used to easily construct the Mesh datastructure of the initial, unit cell (Fig. 9b-6, 7, 8).

The EGI algorithm can operate on an existing cell, or manually provided information about the force vectors. Various properties of the point masses, such as its type (real, zero or virtual) can be input as vertex attributes. The unit cells shown in this paper have faces that are planar to the plane defined by its corresponding normals and the point masses in the EGI. However, the faces of the unit cell



Fig. 7. Face re-sizing with respect to a target area: (a) face partitioned into subtriangles; (b) two new inner edges and the angle β that can be used to determine the area of each sub-triangle; and (c) the new face now with the target area, scaled by factor of *s*.

do not necessarily have to be in the correct target face orientations in the beginning, as the planarisation process is built into the AP algorithm.

3.3. Area pursuit algorithm

.t

Once the topological datastructure of the unit cell has been constructed, the AP algorithm re-sizes each face of the cell towards their target face areas. At each iteration, the faces are re-sized individually and then new vertices are computed for the cell.

Re-sizing a face with a target area can be formulated as a polygon scaling problem. The same technique is implicitly mentioned in [9], but no detailed procedure was provided or explained, especially for non-general cases. Assuming a cell is closed and convex, then all of its constituent faces are polygons that are also closed and convex. The area of face $f_{i,j}$ with n number of edges can be computed by deconstructing the face into n number of subtriangles using the barycentre $\mathbf{b}_{i,j}$ (Fig. 7a).

The area $A_{f_{i,j}}$ of face $f_{i,j}$ is the sum of the areas of all the subtriangles. The area $A_{f_{i,j,n}}$ of the *n*th sub-triangle $f_{i,j,n}$ can be computed using simple trigonometry:

$$A_{f_{i,j,n}} = q \cdot A_{f_{i,j}} = \frac{1}{2} \cdot |\mathbf{r}_n| \cdot |\mathbf{r}_{n+1}| \cdot \sin\beta$$
(1)

where *q* is the ratio of the area of the *n*th sub-triangle $A_{f_{i,j,n}}$ to the area of the total area $A_{f_{i,j}}$ of face $f_{i,j}$, and β is the angle between \mathbf{r}_n and \mathbf{r}_{n+1} (Fig. 7b). If the face $f_{i,j}$ is scaled by the factor of *s* from $\mathbf{b}_{i,j}$ to satisfy the target face area of $A_{f_{i,j}}^t$, the new area of the *n*th sub-triangle $f_{i,j,n}$ is:

$$A_{f_{i,j,n}} = \mathbf{q} \cdot A_{f_{i,j}}$$

$$= \frac{1}{2} \cdot |\mathbf{r}_n \cdot \mathbf{s}| \cdot |\mathbf{r}_{n+1} \cdot \mathbf{s}| \cdot \sin \beta$$

$$= \frac{1}{2} \cdot s^2 \cdot |\mathbf{r}_n| \cdot |\mathbf{r}_{n+1}| \cdot \sin \beta.$$
(2)

Solving for q from Eqs. (1) and (2), the scale factor can be obtained as:

$$s = \sqrt{\frac{A_{f_{i,j}}^t}{A_{f_{i,j}}}}.$$
(3)

With this factor *s*, each face can be scaled such that the new face area is equal to the target area. At any point during the iteration, faces may become self-intersecting, or *complex* [9] (Fig. 8a). The area of a complex face may be found by splitting the edges of the face wherever there is a self-intersection (Fig. 8b). Keeping the original directions of the parent edges, all sub-faces now have split edge cycles in either clockwise or counter clockwise directions (Fig. 8c). Using the split edge cycle directions of the sub-faces, the normal directions of the sub-faces can be determined using the



Fig. 8. (a) A complex face with self-intersecting edges; (b) edges split at the self-intersections, while maintaining the parent edge directions; (c) areas of the sub-faces with the normal directions taken into account; (d) face collapsing; and (e) face untangling.

right-hand rule, and the area of the face is the sum of the subface areas, with its normals determining whether it contributes positively or negatively.

However, determining the scale factor *s* for a complex face is not straightforward since some sub-faces contribute negatively towards the total face area. Because it is assumed that the eventual cell is convex and has no faces that are complex in its final state, it is unnecessary to have complex faces present at any point during the iteration. For a face that has become complex, its vertices can either be collapsed or untangled. If the target face area is 0, the face vertices can collapse towards the closest self-intersection (Fig. 8d). If the target face area is greater than 0, the face vertices of negative sub-faces can be untangled (Fig. 8e).

After re-sizing, collapsing or untangling (Fig. 9c-9,10), the new face vertices can be projected onto the target plane defined by its target normal and its current centroid. The target plane can also be redefined for each iteration as the best-fit plane from the new vertex coordinates (Fig. 9c-11). Allowing faces to be adaptive to the new target planes is an essential feature that will enable Ψ^{\perp} to adjust its face orientations to satisfy the given force-constraints and therefore output a new form-found geometry of the structure. This feature will be demonstrated in Section 4.4. Because each face is scaled independently, each iteration results in multiple coordinates for each of the cell vertices (Fig. 9c-12). The new coordinates for the cell vertices can be computed by averaging these new coordinates (Fig. 9c-13). Iteration continues until a desired tolerance has been reached for all of the cell faces (Fig. 9c-14).

3.4. Datastructure

Previous computational implementations of 3D graphic statics have been based on either the winged-edge [2] or the VolMesh [4] datastructures. These datastructures are ideal for top-down design workflows like global subdivision and transformation operations (Fig. 10a). In a Ψ^{\perp} , adjacent cells are detached and do not necessarily have matching contact faces. Therefore, a single volumetric mesh cannot be used to represent the datastructure of Ψ^{\perp} .

A new hybrid datastructure is proposed, where a Mesh datastructure represents the cell of each node, and a Network datastructure represents the assembly of those individual cells. This datastructure is implemented as *Network of disjointed force polyhedra* (NDFP) using the Network and Mesh classes of the *COMPAS* library [30]. The EGI algorithm runs in the background to ensure





Fig. 9. Condensed Python code snippets for: (a) the EGI algorithm; (b) function for constructing the cell Mesh datastructure from the EGI; and (c) the AP algorithm.



Fig. 10. (a) A typical winged-edge or VolMesh datastructure used in 3D graphic statics; and (b) a NDFP datastructure, with EGI data and cell geometry embedded into each node.

that the topology and geometry of the individual cells are correct and up to date. The AP algorithm ensures that all pairs of adjacent cells have contact faces with equal areas. The \perp algorithm, enforces perpendicularity between Ψ and Ψ^{\perp} .

3.5. Visualisation

Although one of the most valuable benefits of computational graphic statics is the visualisation and the explicit control of both the structure's geometry and its equilibrium of forces, the form and force diagrams increasingly become visually cluttered and illegible as structures become more complicated. The illegibility is even more severe for polyhedral reciprocal diagrams, where it is quite difficult to perceive quantitative information through volumes and face areas of solid geometries [33], especially when they are represented as 2D images on flat media.

Therefore, the visualisation of Γ^{\perp} and Ψ^{\perp} needs to be improved in order to fully take advantage of the inherent benefits of graphic statics, and make the polyhedral reciprocal diagrams more legible, useable and interactive.

3.5.1. Unified diagram

The unified diagram, $\Gamma^{\perp}(\alpha)$, represents both the geometry and internal forces of a structure in a single diagram, thereby improving the legibility of reciprocal diagrams [34] (Fig. 11). Contributions by McRobie have extended this unified diagram concept to 3D structures using the "Minkowski Sum" diagrams, and has shown its numerous benefits in providing new insights and deeper mathematical explanations behind the unified diagrams [35,11]. By parametrically modifying the scaling factor α , all cells of $\Gamma^{\perp}(\alpha)$ are scaled relative to its corresponding nodes of the structure such that the distance between any pair of adjacent cells is $\alpha \cdot L$, where *L* is the length of the corresponding member in Γ .

A scaling factor of 1 results in Γ , whereas lower values of α closer to 0 will result in a $\Gamma^{\perp}(\alpha)$ that more closely resembles



Fig. 11. (a) A simple structure in equilibrium; and (b) the $\Gamma^{\perp}(\alpha)$ with varying scaling factor of α .



Fig. 12. (a) Interstitial prism between two cells of a $\Gamma^{\perp}(\alpha)$; (b) the collapsed cell and the two interstitial prisms between two cells of a $\Psi^{\perp}(\alpha)$; and (c, d) the convex hull as a visual alternative to (b) with the collapsed cell omitted for two cells with the same valency, and two cells with different valencies.

the polyhedral force diagram Γ^{\perp} (Fig. 11b). The volume of the interstitial *prisms* that are formed in between the adjacent cells is equivalent to the work $\mathbf{f} \cdot L$ being done by the corresponding member, where \mathbf{f} is the internal force in the member between the two nodes, and *L* is the length of that member (Fig. 12a).

 $\Gamma^{\perp}(\alpha)$ is not only more discernible, but also provides an interesting visual representation of the material required for a uniform stress design [35]. The unified diagram reveals visual insights in relation to some of the most fundamental principles of structural engineering and analysis, such as: kinematics and mechanisms [34]; virtual work and displacements [36]; and stress-fields and strutand-tie models [37,38].

3.5.2. Unified diagram algorithm

For a Ψ^{\perp} , construction of the unified diagram, $\Psi^{\perp}(\alpha)$, is not as straightforward since the prisms cannot simply be extruded due to the mismatching contact faces. A true $\Psi^{\perp}(\alpha)$ would show the collapsed cells between two disjointed cells, with two prisms instead of one (Fig. 12b). However, with the priority being placed on maximising visual clarity and legibility of the diagrams, the representation of the two prisms and a collapsed cell can be simplified by using a convex hull of the two contact faces (Fig. 12c–d). Note that the volume of this convex hull is not **f** · *L*.

The UD algorithm re-sizes and relocates the cells of Ψ^{\perp} to their correct locations relative to one another. Once the Ψ and Ψ^{\perp} are in their final, perpendicularised states, the scale factor α and the lengths of the corresponding members can be used to determine the distance between any two adjacent cells in $\Psi^{\perp}(\alpha)$. The relative positions between the cells are iteratively computed.

It is important to note here that the UD algorithm is applied purely as a visual approximation to improve the legibility of Ψ^{\perp} . A



Fig. 13. Simple examples – force vectors in equilibrium, the EGI, the unit cell and the final geometry of the polyhedron or cell (from left to right): (a) rectangular box; (b) pentagonal pyramid; (c) tetradecahedron; (d) pentagonal trapezohedron; (e) dodecahedron; and (f) irregular, asymmetric polyhedron.



Fig. 14. (a) Γ of a Jessen icosahedral tensegrity structure; (b) the complete $\Gamma^{\perp}(\alpha)$ using complex cells and "zero bars" (after [42]); and (c) $\Psi^{\perp}(\alpha)$, without the use of any complex cells, "zero-volume cells" or "zero bars".



Fig. 15. A twisting arch bridge with non-planar faces, and the corresponding $\Psi^{\perp}(\alpha)$.

 Γ^{\perp} for a structure having both compression and tension elements typically consist of cells that are both outward and inward in their *cell directions*; the face normals of some cells point inward, and some cells outward [4,9]. In this paper, all cells of a Ψ^{\perp} are represented as inward or outward. Therefore, the distance between any two neighbouring cells as a result of the UD algorithm is not necessarily precisely $\alpha \cdot L$. Although the cell directions are inherently embedded in the winged-edge or VolMesh datastructures, determining the correct cell directions in a network of equilibrated cells that are disjointed is not trivial, and not addressed in this paper.

4. Results

This section presents examples that demonstrate new structural design possibilities using disjointed force polyhedra.

All structures presented are pin-jointed spatial trusses carrying axial forces only. Fully pinned nodes are shown as circles outlined in black, and partially restrained nodes (explained per example) are shown as dotted circles. Applied loads are represented in green. In the force diagrams, cell faces in red correspond to tensile members, and in blue or white correspond to compression members. Interstitial prisms that are represented in pink correspond to members that are in tension, and in light blue correspond to members in compression.

4.1. Simple examples

In order to validate the presented methods, EGI and AP algorithms are applied to reconstruct known polyhedral geometries in literature and an irregular polyhedron, only from its face normals and areas. In Fig. 13, the first column shows the equilibrated force vectors as spike models. The second column shows the corresponding EGIs with adjacency and cross-adjacency arcs. The third column shows the unit cell with zero faces highlighted in orange. Finally, the last column shows the final geometry of the polyhedron with face areas that match the magnitudes of the corresponding force vectors.

4.2. Improved visualisation for 3D graphic statics

Some structures have extremely complicated force diagrams with complex faces and cells. While necessary for constructing a complete Γ^{\perp} , such overlapping elements only make Γ^{\perp} more difficult to read and understand. For a designer who is interested in exploring the design space using Γ^{\perp} , these additional "zero-volume cells" and "zero-bars" serve little purpose.

Complicated self-stressed structures, such as the Jessen icosahedral tensegrity (Fig. 14a), are used commonly in literature to demonstrate the need for zero-volume cells and zero bars to construct a complete Γ^{\perp} (Fig. 14b) [10,35,27]. All of the complex faces of the central, "zero-volume cell" highlighted in Fig. 14b have areas of zero, and therefore the cell has a volume of zero. The geometric properties of the Jessen icosahedron is well known [39–41], and since tensegrity structures are self-stressed and self-equilibrated, the construction of Ψ^{\perp} is simple and straightforward once the vertex locations have been determined. $\Psi^{\perp}(\alpha)$ shown in Fig. 14c is drawn with the same α as the $\Gamma^{\perp}(\alpha)$ in Fig. 14b, but without the "zero-volume cells" and "zero bars" and therefore reducing significant amount of visual clutter.

With the equilibrium of the external loads and reaction forces always being verified by Λ^{\perp} and the individual cells being generated and visualised per node-by-node basis, Ψ^{\perp} can be constructed for any structure in static equilibrium without using any complex



Fig. 16. A layered and self-overlapping shell structure supporting a flat surface that is uniformly loaded.



Fig. 17. A spatial tree structure with non-polyhedral sub-spaces, as a result of node-by-node transformations of Ψ^{\perp} .

faces, complex cells or any additional fictitious nodes or prisms. This example shows that $\Psi^{\perp}(\alpha)$ can be used as a simplified and improved visualisation alternative to a complete but more complicated $\Gamma^{\perp}(\alpha)$.

4.3. New structural typologies

In this section, we present several new structural typologies for which a Ψ^{\perp} can be constructed and used for design.

4.3.1. 2D and 3D combined structures

Using prism-like cells with virtual faces to represent 2D nodes in equilibrium as described in Section 3.2, Ψ^{\perp} for structures with both 2D and 3D nodes can now be constructed. Fig. 15 is a twisting arch bridge, with 2D nodes along the arch, and point loads applied to the deck. Combination of 2D and 3D nodes in the same structure means that the members that form a face of Ψ are not necessarily planar, and thereby allows incorporation of twisted faces and features into the structure.

4.3.2. Overlapping structures

A complete Γ^{\perp} can be constructed only if Γ is a planar graph [43]. This is because there does not exist a topological dual of a graph that cannot be redrawn in the plane without crossing edges, or untangled in space without any self-overlapping edges.

The example in Fig. 16 is a layered shell structure with overlapping vertical support elements. Although this structure is in equilibrium, the complete Γ^{\perp} cannot be constructed. However, through a network of disjointed cells that are individually in equilibrium, a Ψ^{\perp} can be constructed.

4.3.3. Non-polyhedral structures

The dual and reciprocal relationship between Γ and Γ^{\perp} means that both diagrams are polyhedral in their geometric properties. Subsequently, any structure generated through subdivisions or transformations of Γ^{\perp} has sub-spaces that are also polyhedral.

The spatial tree structure shown in Fig. 17 was generated by node-by-node transformations, which result in an equilibrated and



Fig. 18. Overview of the main steps of the force-driven design workflow using the NDFP datastructure, and the four algorithms (\perp , EGI, AP, UD).

yet non-polyhedral form diagram, Ψ . Ψ^{\perp} allows exploration of equilibrated structures that do not have polyhedral geometries, and investigation of free-form designs that are more organic in their aesthetic is now possible. Furthermore, any force equilibrium of structures generated with other form-finding methods such as FDM and TNA can be translated into Ψ^{\perp} .

4.4. Force-driven design

This section presents applications of disjointed force polyhedra where the faces are adaptive, and are now allowed to re-orient themselves at each iteration. Applied in conjunction with the \perp algorithm to a larger network of cells, the adaptive faces enable force-constrained form finding and design explorations using 3D graphic statics.

4.4.1. Workflow

While polyhedral transformations of Γ^{\perp} are ideal for initial form generation and explorations, Ψ^{\perp} provides a means for designers to interactively incorporate force-driven constraints. A typical design workflow using both of these design processes is as follows.

First, boundary conditions of the design problem are clearly identified by the designer, from which the initial Λ^{\perp} can be generated using the EGI and the AP algorithms (Fig. 18a).

The designer then can proceed to subdividing or transforming the Λ^{\perp} (Fig. 18c). Using the \perp algorithm, the designer can interactively generate Γ , and explore the geometry of the structure in real-time (Fig. 18d). During this interactive exploration, the designer can set various form-driven constraints such as node locations, fixities, edge lengths and orientations (Fig. 18b).

Once a general form and topology of Γ have been determined, the designer can proceed to disjointing the Γ^{\perp} using the EGI and AP algorithms (Fig. 18e), which subsequently converts Γ to Ψ . Because the force in every member is already known, the construction of the initial Ψ^{\perp} is straightforward. The designer can now set various force-driven constraints such as target force magnitudes or orientations for specific members of the structure (Fig. 18f). At this point, the designer can still apply any form-driven constraints to Γ , or polyhedral transformations to Ψ^{\perp} .

Because Ψ and Ψ^{\perp} have an interdependent relationship, the geometry of Ψ needs to be updated at each iteration as each of the cells adapts to the new force constraints (Fig. 18h). At the end of each iteration, the two corresponding contact faces of a member may not necessarily be parallel or have the same areas. For members that do not have target member forces or orientations, the average of the two contact face normals and areas are used as targets for the next iteration.

The iteration is terminated when the desired tolerance or a designated iteration count has been reached. The edges of Ψ and the corresponding cell faces of Ψ^{\perp} should now be close to being perpendicular to one another, unless the form and force constraints input by the designer in the previous steps caused the polyhedral reciprocal diagrams to be over-constrained and a solution satisfying all of the constraints could not be found. In this case, the designer will need to consider eliminating some of the constraints.

As the final step, the designer has the option to visualise the unified diagram, $\Psi^{\perp}(\alpha)$, which aids in understanding the force magnitudes and distributions relative to Ψ (Fig. 18i). The designer can also go back to the previous steps to continue the design exploration.

4.4.2. Construction of global force polyhedron

The resolution of the resultant force and the construction of the global force polygon or polyhedron are some of the most fundamental principles of graphic statics. By constructing the global force polyhedron, Λ^{\perp} , the equilibrium of the external loads and reaction forces is established [25,26,44]. This means that any polyhedral subdivisions or transformations of Λ^{\perp} results in a structure that is also internally equilibrated. Because the internal equilibrium of the structure is guaranteed, a wide range of structures can be explored very rapidly. The EGI and AP algorithms allow quantitative boundary condition constraints to be incorporated during the construction of Λ^{\perp} .

Suppose that a shell structure with five supports is considered for a hypothetical site shown in Fig. 19a. The client has requested for two of the supports to be on piers 1 and 4. For the first iteration, a solution was found where all of the supports had the same magnitude of reaction forces (Fig. 19b). However, it became evident that the two piers may be susceptible to high horizontal thrusts. By limiting the horizontal thrusts on piers 1 and 4 to not exceed a certain amount, a much shallower Λ^{\perp} was found (Fig. 19c). As a



Fig. 19. (a) A hypothetical site for a shell-like structure with five supports; (b) Λ^{\perp} where all the reaction force magnitudes are equal; (c) Λ^{\perp} where all the reaction force magnitudes are equal, but the horizontal components of the reactions at piers 1 and 4 are limited to stay below a desired amount; and (d) imposing constraints from both (b) and (c), while keeping all of the supports clear of the water front area.



Fig. 20. (a) An indeterminate truss with four horizontally restrained supports (1–4) and one pinned support (5), and the corresponding $\Gamma^{\perp}(\alpha)$; and (b) one possible equilibrium solution for the same truss with a point load applied to the interior node, and the corresponding Ψ and $\Psi^{\perp}(\alpha)$.

last constraint, the client requests that none of the supports land on the water front area. By constraining the supports 2, 3, and 5 to stay clear of this zone marked by the red lines shown in Fig. 19d, while satisfying the force constraints imposed in Fig. 19a and b, a new Λ^{\perp} is found.

Once a Λ^{\perp} has been found that meets the main boundary condition criteria, the designer can then proceed to applying various subdivision and transformations to Λ^{\perp} to explore more specific forms and topologies of the structure, knowing that the required boundary condition criteria have already been satisfied.

4.4.3. Placing point loads anywhere

One of the main limitations of previous graphic statics applications is that the external loads must be applied at the periphery of the structure, meaning there cannot be any "inner leaves" [43]. This is also true for 3D graphic statics, where any inner leaves or crossing members mean that a topological dual does not exist. With Ψ^{\perp} , point loads can be placed anywhere in the structure in any direction.

Fig. 20a is an indeterminate truss with four horizontally restrained supports (1–4) and one pinned support at the bottom (5). If a point load were to be applied to the inner, central node of the structure, a complete Γ^{\perp} cannot be constructed as there does not exist a topological dual for such configuration of edges.

However, in Fig. 20b, a point load of 0.25*P* is applied to the central node at an arbitrary angle, from which a new Ψ and Ψ^{\perp} were found. The new added freedom to place loads anywhere in the structure, allows investigation of irregular loading scenarios, asymmetric loading conditions, and potentially incorporate self-weight loads.

4.4.4. Tributary area

While polyhedral subdivisions and transformations allow generation of intricate spatial structures, the resulting distribution of applied loads on the structure does not end up representing realistic loading scenarios.

Consider the spatial tree structure shown in Γ of Fig. 21a, which is designed to support a triangle-shaped roof that weighs *P*. Suppose that the points on top and the bases of the structures are finalised by the design team, and are fixed for the remainder of the design exploration. As shown in Γ of Fig. 21a, the distribution of applied forces as a result of polyhedral transformations of Γ^{\perp} often do not correctly reflect the true tributary areas of the structure. With the top and the base points fixed, and the correct distribution of applied loads imposed, the new shape of the design is found (Fig. 21b).

A designer is typically concerned with rapid shape explorations during early stages of design. As the design is gradually finalised and a specific topology of the structure is chosen, the designer can begin to adjust the design load case of the initial Γ^{\perp} to a more correctly calibrated loading scenario to continue developing the design towards a more realistic version of the initial concept.

4.4.5. Interactive force-driven design

Implementation of the proposed method in an interactive modelling environment allows designers to explore various spatial structures based on specific force-driven constraints in real-time.

Consider a vertically loaded column in Fig. 22a, with a fully pinned support at the bottom and a horizontally restrained support at the top. Any combination of edges in Ψ or cell faces of



Fig. 21. (a) A tree structure generated through polyhedral subdivisions and transformations of Γ^{\perp} , with a distribution of applied loads that do not reflect the correct tributary areas; and (b) the same structure with correct distribution of applied loads according to the actual tributary areas, and the subsequently form-found, new tree structure.

 Ψ^{\perp} can be selected to input specific target force magnitudes. In general, Ψ for a given force distribution is not always unique, and is subject to certain geometric constraints such as maximum and minimum length of edges allowed, and node location constraints during the form-finding process. Consequently, a design problem can become over-constrained and an equilibrium solution may not be found that satisfies all of the input constraints. In such cases, the converged solution then provides the designer with the closest solution given the input constraints, and indicates where certain constraints can be removed or modified. The presented framework allows designers to interactively set different combinations of constraints to explore various equilibrium solutions.

4.4.6. Synthesising polyhedral methods

This last example demonstrates the full potential of forcedriven structural design using disjointed force polyhedra.

Suppose that a design for a roof structure generated through polyhedral subdivisions is being considered (Fig. 23). Through polyhedral transformations of Γ^{\perp} , various equilibrium structures can be explored, while the faces of Γ are inherently constrained

to be planar. This built-in planarity property of Γ is ideal for both fabrication and construction of the geometry without the need of any additional optimisation processes.

The roof will be placed on six existing columns, therefore no horizontal reactions are allowed at the base of the roof. Rather than using straight cable ties to counterbalance the horizontal thrusts, the designers want to explore a more integrated cablenet-like design to maximise the visual and spatial experience from below. Furthermore, the designers want to consider using two perimeter cables that have constant force throughout their length. Constant force members in trusses and various other structures are beneficial in that it allows a single cross section to be used, and the material is utilised to its full capacity throughout its length [6].

In 2D graphic statics, the construction of constant-force trusses is intuitive and can be completed procedurally using a circle as a geometric constraint on the force diagram. However, in 3D graphic statics, such geometric constraint is not easy to define or impose. An optimisation technique is mentioned and applied to an example in [9, p. 148–150]. As the author notes, however, optimising the face areas of a complete Γ^{\perp} may not always converge, and is highly dependent on the initial geometric properties and topologies of J. Lee et al. / Computer-Aided Design 99 (2018) 11-28



Fig. 22. Interactive design explorations of a vertically loaded column using various combinations of target force magnitudes for specific members of the structure. Each example shows: the initial structure with the target force magnitudes; the corresponding $\Psi^{\perp}(\alpha)$; and the form-found Ψ .

the cell faces, since the topologies of the cells are not allowed to change.

Using disjointed force polyhedra, the force distribution in the cable net can be controlled more precisely; the cell faces corresponding to the perimeter cables of the structure are constrained to have the same areas. At the same time, the geometry of the primary compression structure above can be constrained to remain purely

polyhedral with planar faces, which is more ideal for fabricationdriven constraints.

While similar to other force-constrained form-finding methods such as FDM and TNA, the presented force-driven design explorations are based on an exclusively geometric framework with various built-in polyhedral properties. The designers' ability to maintain and impose these built-in polyhedral constraints to either the entire structure or only specific parts of the structure



Fig. 23. Combinatorial application of polyhedral design methods for a roof structure with no horizontal reactions. Transformation of Γ^{\perp} is used to generate the polyhedral geometry of the structure and the outer layer of the roof, where the equilibrium and fabrication constraints are dominant. For the cable net, Ψ^{\perp} is used to enforce a constant-force constraint on the two main cables.

as needed, is a unique feature that is not possible with other methods.

5. Conclusions

This paper presented a computational framework for disjointed force polyhedra, which is an extension of 3D graphic statics that broadens the range of design applications of polyhedral reciprocal diagrams.

Building upon the previous work on polyhedral reconstruction methods from face normals and areas, we presented the robust EGI algorithm that constructs a flexible and versatile datastructure that is more suitable for the purposes of interactive structural design. In conjunction with the EGI algorithm, the AP algorithm is an iterative geometric solver that computes the geometry of the polyhedra with target face normals and areas. These algorithms, with previously presented reciprocation algorithm, were synthesised into a hybrid, NDFP datastructure that allows designers to use disjointed force polyhedra in an interactive way, with the ability to set a variety of geometric and force-driven constraints at any point during the design process. Lastly, the UD algorithm was presented, which extends unified diagram representation method for disjointed force polyhedra.

Through a series of examples in Section 4, we demonstrated how disjointed force polyhedra can be used to simplify the visualisation of polyhedral force diagrams by eliminating complex faces and cells. These examples also showcased new typologies of structures that can now be explored with disjointed force polyhedra. Furthermore, we showed how the computational framework for disjointed force polyhedra can be used independently from, or in combination with existing design techniques of 3D graphic statics and potentially other form-finding methods such as FDM and TNA. Most importantly, these examples presented how polyhedral force diagrams can now address force constraints, which allows incorporation of more realistic structural design and engineering criteria, and enable force-driven graphical method of structural design that was simply not possible with previous implementations of 3D graphic statics.

Acknowledgements

The author would like to thank the post-doctoral researchers of the Block Research Group (Dr. Matthias Rippmann, Dr. Tomás Méndez Echenagucia and Dr. Andrew Liew) for their constant technical support and insightful comments.

References

- Rankine W. Principle of the equilibrium of polyhedral frames. Philos Mag Ser 4 1864;27(180):92.
- [2] Akbarzadeh M, Van Mele T, Block P. On the equilibrium of funicular polyhedral frames and convex polyhedral force diagrams. Comput Aided Des 2015;63:118–28.
- [3] Akbarzadeh M, Van Mele T, Block P. Three-dimensional compression form finding through subdivision. In: Proceedings of the IASS Symposium 2015, Amsterdam, The Netherlands, 2015.
- [4] Lee J, Van Mele T, Block P. Form-finding explorations through geometric transformations and modifications of force polyhedrons. In: Proceedings of the IASS Symposium 2016, Tokyo, Japan, 2016.
- [5] Van Mele T, Lachauer L, Rippmann M, Block P. Geometry-based understanding of structures. J Int Assoc Shell Spatial Struct 2012;53(174):285–95.
- [6] Allen E, Zalewski W. Form and forces: Designing efficient and expressive structures. New York: Wiley; 2009.
- [7] Schek H. The force density method for form finding and computation of general networks. Comput Methods Appl Mech Engrg 1974;3:115–34.
- [8] Block P, Ochsendorf J. Thrust network analysis: A new methodology for threedimensional equilibrium. J Int Assoc Shell Spatial Struct 2007;48(3):167–73.
- [9] Akbarzadeh M. 3D graphical statics using reciprocal polyhedral diagrams (Ph.D. thesis), Zürich, Switzerland: ETH Zürich; 2016.
- [10] McRobie A. Rankine reciprocals with zero bars, 2017. URL https://www.researc hgate.net/publication/315027343_Rankine_Reciprocals_with_Zero_Bars (Accessed: June, 2017).
- [11] McRobie A. The geometry of structural equilibrium. Roy Soc Open Sci 2017; 4(3).
- [12] McRobie A. Graphic analysis of 3d frames: Clifford algebra and rankine incompleteness. In: Proceedings of the IASS Symposium 2017, Hamburg, Germany, 2017.
- [13] Demaine ED, O'Rourke J. Geometric folding algorithms: linkages, origami, polyhedra. Cambridge University Press; 2007. p. 339–57 (Chapter 23).
- [14] Minkowski H. Allgemeine lehrsätze über die convexen polyeder. Nach Ges Wissensch Göttingen Math-Phys Kl 1897;198–219.
- [15] Alexandrov A. Convex polyhedra. 1st ed. Springer monographs in mathematics, Berlin Heidelberg: Springer-Verlag; 2005.
- [16] Grünbaum B. Convex polytopes. 2nd ed. Graduate texts in mathematics, New York, Berlin, London: Springer; 2003.
- [17] O'Rourke J. Convex polyhedra realizing given face areas, CoRR abs/1101.0823, 2011. URL http://arxiv.org/abs/1101.0823 (Accessed: November, 2016).
- [18] Ikeuchi K. Recognition of 3-d objects using the extended gaussian image. In: IJCAI, 1981, p. 595–600.
- [19] Little JJ. An iterative method for reconstructing convex polyhedra from external guassian images. In: AAAI, 1983, p. 247–50.
- [20] Horn BKP. Extended gaussian images. Proc IEEE 1984;72(2):1671-86.
- [21] Moni S. A closed-form solution for the reconstruction of a convex polyhedron from its extended Gaussian image. In: Proceedings of the 10th international conference on pattern recognition, Vol. 1, 1990, p. 223–6.

- [22] Xu Z, Suk M. Representation and reconstruction of polygons and polyhedra using hierarchical extended gaussian images. Ann Math Artif Intell 1995;13(3):377–99.
- [23] Gritzmann P, Hufnagel A. On the algorithmic complexity of Minkowski's reconstruction theorem. J Lond Math Soc 1999;59(2):1081–100.
- [24] Lachand-Robert T, Oudet É. Minimizing within convex bodies using a convex hull method. SIAM J Optim 2005;16(2):368–79.
- [25] Akbarzadeh M, Van Mele T, Block P. 3D Graphic statics: Geometric construction of global equilibrium. In: Proceedings of the IASS Symposium 2015, Amsterdam, The Netherlands, 2015.
- [26] Akbarzadeh M, Mele TV, Block P. Three-dimensional graphic statics: Initial explorations with polyhedral form and force diagrams. Int J Space Struct 2016;31(2-4):217–26.
- [27] Konstantatou M, McRobie A. Reciprocal constructions using conic sections poncelet duality. In: Proceedings of the IASS Symposium 2016, Tokyo, Japan, 2016.
- [28] Mitchell T, Baker W, McRobie A, Mazurek A. Mechanisms and states of selfstress of planar trusses using graphic statics, Part I: The fundamental theorem of linear algebra and the Airy stress function. Int J Space Struct 2016; 31(2-4):85-101.
- [29] McRobie A, Williams CJ. A stress function for 3d frames. Int J Solids Struct 2017;117:104–10.
- [30] Mele TV, Liew A, Echenagucia TM, Rippmann M. et al., COMPAS: A framework for computational research in architecture and structures, 2017. http: //compas-dev.github.io/compas/.
- [31] Python software foundation, Python Language Reference, Version 2.7.11 (2001-2016). URL http://www.python.org.
- [32] Roach J, Wright J. Spherical dual images: A 3d representation method for solid objects that combines dual space and Gaussian spheres. In: Proceedings of the 1986 IEEE international conference on robotics and automation, Vol. 3, 1986, p. 1087–92.
- [33] Cleveland WS, McGill R. Graphical perception: Theory, experimentation, and application to the development of graphical methods. J Amer Statist Assoc 1984;79(387):531–54.
- [34] Zanni G, Pennock GR. A unified graphical approach to the static analysis of axially loaded structures. Mech Mach Theory 2009;44(12):2187–203.
- [35] McRobie A. Maxwell and Rankine reciprocal diagrams via Minkowski sums for two-dimensional and three-dimensional trusses under load. Int J Space Struct 2016;31(2-4):203-16.
- [36] McRobie A, Konstantatou M, Athanasopoulos G, Hannigan L. Graphic kinematics, visual virtual work and elastographics. Roy Soc Open Sci 2017;4(5).
- [37] Schlaich M, Anagnostou G. Stress fields for nodes of strut-and-tie models. J Struct Eng 1990;116(1):13–23.
- [38] Muttoni A, Schwartz J, Thürlimann B. Design of concrete structures with stress fields. Birkhäuser Basel; 1997.
- [39] Jessen B. Orthogonal icosahedron. Nordisk Mat Tidskr 1967(15):90-6.
- [40] Fuller B. Synergetics: explorations in the geometry of thinking. New York: Macmillan; 1975.
- [41] Robert William Burkhardt J. A practical guide to tensegrity design, September 2008. URL http://www.angelfire.com/ma4/bob_wb/tenseg.pdf (Accessed: 02.11.17).
- [42] McRobie A, Baker W, Mitchell T, Konstantatou M. Mechanisms and states of self-stress of planar trusses using graphic statics, Part III: Applications and extensions. In: Proceedings of the IASS Symposium 2015, Amsterdam, The Netherlands, 2015.
- [43] Van Mele T, Block P. Algebraic graph statics. Comput Aided Des 2014;53: 104–16.
- [44] Lee J, Van Mele T, Block P. Area-controlled construction of global force polyhedra. In: Proceedings of the IASS Symposium 2017, Hamburg, Germany, 2017.



Juney Lee is a Ph.D. researcher at the Block Research Group of ETH Zurich in Switzerland. His research is focused on 3D graphic statics using polyhedral reciprocal diagrams, and computational implementation of graphical methods of design and analysis of structures. He studied architecture and structural engineering at the University of California at Berkeley, and received Master of Architecture and Master of Structural Engineering degrees from Massachusetts Institute of Technology (MIT) in 2015.



Tom Van Mele is the Co-director of the Block Research Group at ETH Zürich, where he leads research and development. In 2008, he received his Ph.D. from the Department of Architectural Engineering at the Vrije Universiteit Brussel (VUB) in Belgium. He is the lead developer of COMPAS, an open-source, Python-based computational framework for collaboration and research in architecture, engineering and digital fabrication.



Philippe Block is a Professor and the Co-director of the Block Research Group at ETH Zürich, where he is also the Director of the Swiss National Centre of Competence in Research (NCCR) Digital Fabrication. He studied architecture and structural engineering at the Vrije Universiteit Brussel (VUB) in Belgium, and received his Ph.D. from the Massachusetts Institute of Technology (MIT) in 2009.