A derivative-free optimization-based approach for detecting architectural symmetries from 3D point clouds

Fan Xue[a], Weisheng Lu[a,1,*], Christopher J. Webster[b], Ke Chen[a]

[a]*Department of Real Estate and Construction, The University of Hong Kong, Pokfulam, Hong Kong SAR*

[b]*Faculty of Architecture, The University of Hong Kong, Pokfulam, Hong Kong SAR*

Abstract

Symmetry is ubiquitous in architecture, across both time and place. Automated architectural symmetry detection (ASD) from a data source is not only an intriguing inquiry in its own right, but also a step towards creation of semantically rich building and city information models with applications in architectural design, construction management, heritage conservation, and smart city development. While recent advances in sensing technologies provide inexpensive yet high-quality architectural 3D point clouds, existing methods of ASD from these data sources suffer several weaknesses including noise sensitivity, inaccuracy, and high computational loads. This paper aims to develop a novel derivative-free optimization (DFO)-based approach for effective ASD. It does so by firstly transforming ASD into a nonlinear optimization problem involving architectural regularity and topology. An in-house ODAS (Optimization-based Detection of Architectural Symmetries) approach is then developed to solve the formulated problem using a set of state-of-the-art DFO algorithms. Efficiency, accuracy, and robustness of ODAS are gauged from the experimental results on nine sets of real-life architectural 3D point clouds, with the computational time for ASD from 1.4 million points only 3.7 seconds and increasing in a sheer logarithmic order against the number of points. The contributions of this paper are three-fold. Firstly, formulating ASD as a nonlinear optimization problem constitutes a methodological innovation. Secondly, the provision of up-to-date, open source DFO algorithms allows benchmarking in the future development of free, fast, accurate, and robust approaches for ASD. Thirdly, the ODAS approach can be directly used to develop building and city information models for various value-added applications.

Keywords: Architectural symmetries, symmetry detection, derivative-free optimization, LiDAR, octree based sampling, Building Information Model

1. Introduction

Symmetry is omnipresent in architecture beyond any epochal, functional, and cultural differences (Fletcher and Fletcher, 1905; Steadman, 1983). Types of architectural symmetries include reflection, translation, rotation, uniform scaling, and their combinations. These are not accidental but instead often result from considerations of function, mechanics, economics, manufacture, and aesthetics (Fletcher and Fletcher, 1905; Mitra et al., 2013). Architectural symmetries often comply with strong co-hierarchical correlations such as collinearity, reflection, and perpendicularity (Zhang et al., 2013) to form a symmetry hierarchy (Wang et al., 2011) embodying rich geometric regularities and topological semantics (Lowe, 1987; Xu et al., 2017, 2018; He et al., 2018).

Recent advances in sensing technology such as LiDAR (Light Detection and Ranging) and photogrammetry have given rise to inexpensive but fine 3D point clouds of buildings and cities (Mallet and Bretar, 2009; Tang et al., 2010). Applied to 3D point clouds, architectural symmetry detection (ASD) has the potential to reveal useful geometric fundamentals for understanding the architecture represented. These geometric fundamentals could be further used, *e.g.*, for registering volumetric, online open BIM components (Xue et al., 2018, 2019b), to develop semantically rich building information models (BIMs) (Eastman et al., 2011; Chen et al., 2018). Properly scaled up, BIMs can be used to form 3D digital city models, increasingly referred to as city information models (CIMs) (Toschi et al., 2017). Semantically rich BIMs and CIMs, so-called 'digital twins', can have innovative applications in areas such as architectural and urban design (Chiaradia, 2009; Haunert, 2012), heritage conservation, construction management (Volk et al., 2014), geophysics (Nghiem et al., 1992), remote sensing (Weidner and Förstner, 1995; Wu et al., 2018), transportation (Ferraz et al., 2016), autonomous robotics (Bajcsy et al., 2018), and smart and resilient city development (Birkmann et al., 2016).

However, while researchers have developed approaches for detecting general symmetries, architectural symmetries included, ASD from 3D point clouds of uncontrolled, real-world architecture remains very challenging. Current approaches for ASD suffer several weaknesses including noise sensitivity (Brown, 1983), high computational loads (Berner et al., 2008), and production of inaccurate results (Lipman et al., 2010), as discussed in greater detail later. As a result, the valuable semantic information contained in architectural symmetries is often unrecognized and unexploited in digital BIM and CIM creation (Van Kaick et al., 2011; Volk et al., 2014; Chen et al., 2018).

This paper proposes a novel derivative-free optimization (DFO)-based approach for fast, accurate, and robust ASD from 3D point clouds. DFO is a subclass of mathematical optimization which has been successfully applied to many challenging problems in science and engineering (Conn et al., 2009), including protein structure prediction (Nicosia and Stracquadanio, 2008), aircraft wing design (Lee et al., 2008), and design of new materials (Miskin and Jaeger, 2013). In this paper, seven state-of-the-art DFO algorithms are selected and benchmarked for ASD in various real-world architectural styles. This is the first research gauging DFO algorithms systematically for ASD, to the best of our knowledge.

2. Existing approaches to detection of symmetries from 3D point clouds

Over the years, researchers have developed different approaches for detecting general symmetries, including architectural symmetries, from 3D point clouds. Xue et al. (2019a) suggested three categories according to methodology: (i) pairwise voting-clustering, (ii) heuristic feature matching, and (iii) parameter optimization. A summary comparison of these approaches in terms of methodology, accuracy, efficiency, and compatible symmetry types is contained in Table 1. In general, heuristic feature matching methods are faster, but pairwise voting-clustering and parameter optimization methods have greater accuracy and compatibility with different types of symmetries (Schnabel et al., 2007; De Berg et al., 2008; Szeliski, 2010; Xue et al., 2019a).

Table 1: Three categories of general symmetry detection method for point clouds

Type of approach	Methodology	Accuracy (less geometric error)	Efficiency (using less time)	Types of symmetries
1. Pairwise voting-clustering	Collection of pairwise votes of *all* the points in the parameter space	++	−	All (++)
2. Heuristic feature matching	Matching features (*E.g.*, lines, planes, spheres) to infer symmetries	−	++	Limited (−)
3. Parameter optimization	Solving abstracted optimization models over the parameter space	++	+	All (++)

++: Very satisfactory; +: satisfactory; −: not satisfactory.

Pairwise voting-clustering methods focus on the correspondence, primarily based on the Hough-like transformation parameter space (Hough, 1959). Given a cloud of n points, the core technology of pairwise voting-clustering is the collection of 'votes' from the $O(n^2)$ combinatorial pairs of the points in the parameter space, where the most voted parameter settings approximately represent the symmetries. In developing this approach, Mitra et al. (2006) first applied Atallah (1985)'s voting-clustering to 3D point clouds, later extending it to architectural models and symmetrization based design (Mitra and Pauly, 2008). Recent advances include detection of dense local symmetry correspondences based on the subspace symmetry (Alexander et al., 2011), the use of wavelet convolution instead of points (Cicconet et al., 2017), and Lie-algebra voting with improved robustness in arbitrary 3D point clouds (Shi et al., 2016). However, voting-clustering approaches have limitations in processing the point clouds of real-life architecture, mainly owing to the inherited bias and proneness to noise of the Hough-like transform (Brown, 1983), ineffective recognition of local symmetries due to loss of correlation information during voting (Bokeloh et al., 2009), low efficiency (exponential to the number of parameters), and unaffordable time consumption given point clouds consisting of millions of points (Berner et al., 2008).

Heuristic feature matching methods detect symmetries indirectly by matching a collection of local geometric features through heuristic rules (or pre-trained models). From a line, plane, sphere, or other such geometric feature can naturally be inferred the locations and tilts of its dual symmetries; a planar rectangle, for example, naturally has two reflection symmetries. Examples of this approach include matching of feature points (Berner et al., 2008), feature lines and their repetitive patterns (Bokeloh et al., 2009; Lin et al., 2015), boundary-tracing (Sampath and Shan, 2007), and other pre-defined features (Forsyth and Ponce, 2012; Schnabel et al., 2007). GlobFit (Li et al., 2011) and RAPter (Monszpart et al., 2015) apply preliminary architectural regularities, for example, emphasizing the adjacent planes with angles of 90° and 60° on building envelopes. While heuristic feature matching approaches are efficient in handling a large number of local symmetries with arbitrary structures (Kerber et al., 2012), they are by their nature primarily confined to the heuristic rules of the point clouds, and require an abundance of suitable features (Lipman et al., 2010).

Parameter optimization methods detect symmetries by perturbing the parameters for the optimal symmetry conditions, such as the maximal point correspondence or the minimal root-mean-square distance (RMSD). For example, Kazhdan (2007) exploits fast Fourier transform (FFT) and formalizes the optimal axis of rotational symmetry; Raviv et al. (2010) present a parameter optimization method for non-rigid 3D objects in general; Wang et al. (2016) detects repetition on facades using an energy optimization function; and Ecins et al. (2017) applied optimal difference on normal directions to detect symmetric objects in indoor point clouds. Some recent heuristic matching methods, such as Globfit (Li et al., 2011) and RAPter (Monszpart et al., 2015), incorporate parameter optimization in part. However, many mathematical methods that guarantee ideal symmetry (or optimality) cannot be applied to 3D point clouds of architectures due to the nonlinear and expensive (time-consuming) symmetry conditions. In addition, the results of ASD can be sensitive to the optimization algorithm's settings.

DFO is a class of mathematical optimization (Conn et al., 2009). Although the derivatives of an objective function contain information vital to finding the best values, such derivatives in many complex science and engineering contexts such as molecular biology, material sciences, engineering modeling, as well as parameter optimization for ASD, are often unavailable, unreliable, or impractical to obtain (Rios and Sahinidis, 2013; Xue et al., 2018). In such circumstances, DFO algorithms incorporate various meta-models of the search space (*e.g.*, the parameter space of symmetries) to carry out optimization. Examples of DFO algorithms are variants of DIviding RECTangles (DIRECT) (Jones et al., 1993; Rios and Sahinidis, 2013), and Covariance Matrix Adaptation Evolution Strategy (CMAES) (Hansen and Ostermeier, 2001; Hansen, 2016), and their applications include protein structure prediction (Nicosia and Stracquadanio, 2008), evolutionary aircraft wing design (Lee et al., 2008)), and optimal adaptation of new materials (Miskin and Jaeger, 2013). Xue et al. (2019a) applied CMAES to a building rooftop point cloud, detecting the global reflection in about 100 seconds; however, this application was still too time-consuming and failed to gauge the CMAES thoroughly and to test other DFO algorithms. Xue et al. (2018, 2019b) also applied CMAES to the processing of low-resolution 2D images to reconstruct the outdoor scene of a demolished building as well as a noisy indoor 3D point cloud for an indoor furniture scene, at an RMSD of 3.87 cm in 6.44 seconds; though, the 3D models of furniture were pre-determined for the test scenes. In summary, DFO algorithms have shown strengths in the solution of challenging and expensive optimization problems (Conn et al., 2009), and thus have great potential in solving the parameter optimization problem of ASD.

3. The proposed Optimization-based Detection of Architectural Symmetries

3.1. Problem formulation for architectural symmetry detection

Symmetry is an affine transformation that preserves points, straight lines, and planes on the 3D Euclidean space ($\mathbb{R}^3$) (De Berg et al., 2008). General symmetry detection is the process of finding the symmetry group G of a 3D point cloud C (James, 1978):

$$\begin{aligned} G &= \langle \mathcal{T}, \circ \rangle, \\ \mathcal{T} &= \{T | T(C) = C, T \text{ is affine on } \mathbb{R}^3\}, \\ C &= \{p_1, p_2, \ldots, p_n\} \subset \mathbb{R}^3, n > 0, \end{aligned} \tag{1}$$

where G is the symmetry group, $\mathcal{T}$ is the set of perfect global symmetries, $\circ$ is the function composition defined on $\mathcal{T}$, *i.e.*, $(g \circ f)(x) = g(f(x))$, C is the the 3D point cloud, and n is the cardinality (number of points) of C. The perfect global symmetry condition is $T(C) = C$ in Eq. 1 (Mitra et al., 2006), *i.e.*, a perfect global symmetry keeps C invariant as a whole while transforming every point. Alternatively, the perfect local (or partial) symmetry condition is that exists $C' \subset$

C such that $T(C') = C'$ (Mitra and Pauly, 2008). In the point clouds of real-life architecture, there are inevitably instrumental, environmental, and calibration errors. Thus, the perfect symmetry conditions are often relaxed to approximate descriptors (Kazhdan, 2007; Mitra and Pauly, 2008), *e.g.*,

$$d(p, T, C) = \|T(p) - N(T(p), C)\|, \tag{2}$$

$$PCR = \frac{1}{n}|\{p|p \in C, d(p, T, C) < \varepsilon \cdot diag_C\}|, \tag{3}$$

$$RMSD = \sqrt{\frac{1}{n}\sum_{p \in C} d(p, T, C)^2}, \tag{4}$$

where $N(p, C)$ in Eq. 2 denotes the nearest point of a point p in C, $d(p, T, C)$ measures the distance of p to C after a transform T, $0 < \varepsilon \ll 1$ is a small error tolerance, and $diag_C$ is the diagonal length of the bounding box of C. Eq. 3 describes the point correspondence rate (PCR) including those points that are still very close to C after a transform T; Eq. 4 indicates the well-known RMSD metric of the whole cloud C after a transform T. If the symmetry plane or axis is known perpendicular to the ground, Eq. 2 can be facilitated with horizontal slices of C or equivalent indices on the z-axis (Xue et al., 2019a). In contrast, Eq. 3 is more fundamental in geometry, while Eq. 4 and its variants such as mean-square error is also effective for parameter optimization methods, such as in the pilot case in Xue et al. (2019a).

An octree is a well-known data structure for indexing and sampling large-scale (*e.g.*, up to one billion points) 3D point clouds (Meagher, 1982; Elseberg et al., 2013). Because the best time complexity of computing Eq. 2 is $O(\log n)$ using the up-to-date $k$dtree-based search, the time complexity of computing Eqs. 3 and 4 is thus $O(n \log n)$, which is still too high for large-scale 3D point clouds.

This paper employs the octree structure for sampling a subset of weighted feature points $C_v \subseteq C$, as demonstrated in Fig. 1, where each point p is associated with a voxel (leaf node) of the octree and a weight w_p representing the number of points in the same voxel, thus $\Sigma_{p \in C_v} w_p = n$. Whenever the maximum octree depth δ increases by 1, the space of each voxel is evenly divided to 8 (*i.e.*, $2 \times 2 \times 2$) new voxels, so that the diagonal $diag_v \propto 0.5^\delta$ (Yamaguchi et al., 1984). Given a large-scale point cloud C of a constant surface area (*i.e.*, very dense points), the cardinality of C_v is inversely proportional to the average cross-sectional area of the voxels, *i.e.*,

$$|C_v| \propto 1/diag_v^2 \propto 4^\delta, \qquad \delta \gg 1. \tag{5}$$

The experimental results of a test case in Fig. 1 confirmed that the cardinality obeyed an exponential equation

$$|C_v| = 4.8429 \times 3.8478^\delta, \qquad R^2 = 0.9947, \tag{6}$$

which validated the theoretical deduction based on the satisfactory R^2, though 3.8478 was slightly less than 4 due to the point density. Therefore, $|C_v|$ is bounded at $O(4^\delta)$ in general, and Eqs. 3 and 4 can thus be approximated in $O(4^\delta \log n)$, respectively:

$$PCR' = \frac{1}{n} \sum_{p \in C_v} \begin{cases} w_p, & \text{if } d(p, T, C) < \varepsilon \cdot diag_C \\ 0, & otherwise \end{cases} \tag{7}$$

$$RMSD' = \sqrt{\frac{1}{n} \sum_{p \in C_v} w_p \cdot d(p, T, C)^2} \tag{8}$$

Based on visual inspections of Fig. 1, a depth of 4 to 6 can approximately represent the surface of an architecture, where the remaining points, *i.e.*, $C \setminus C_v$, can be confidently estimated by linear interpolation using the points and their weights in C_v.

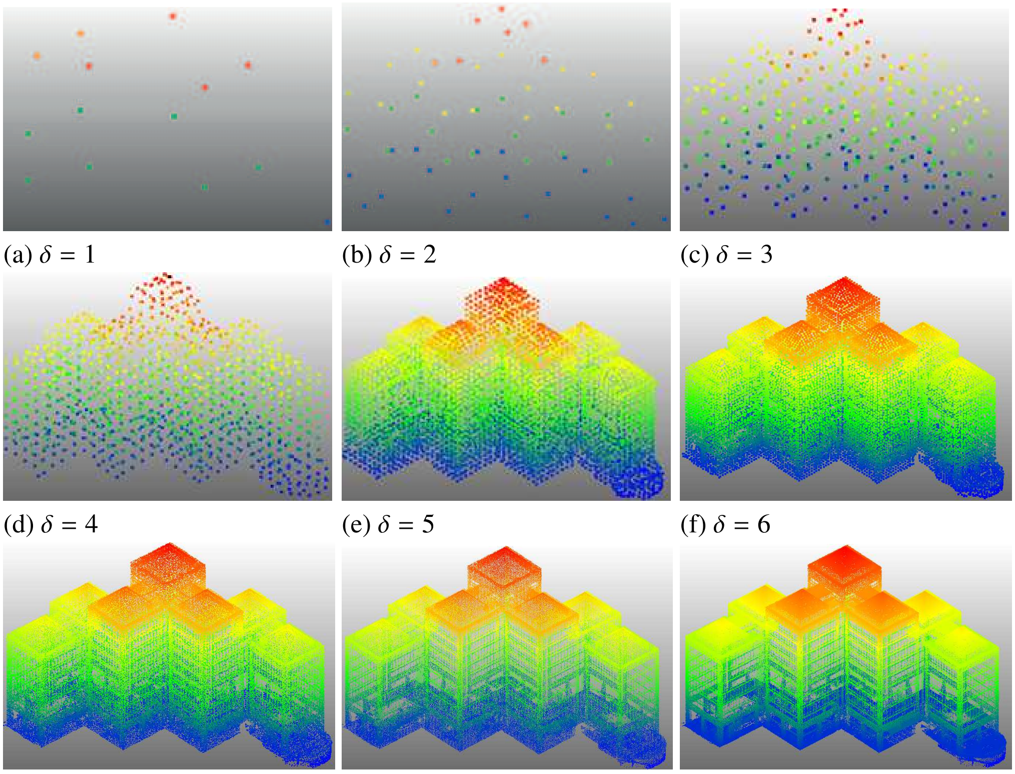

(a) $\delta = 1$ (b) $\delta = 2$ (c) $\delta = 3$
(d) $\delta = 4$ (e) $\delta = 5$ (f) $\delta = 6$
(g) $\delta = 7$ (h) $\delta = 8$ (i) $\delta = 9$

Figure 1: Octree-based weighted feature point sampling for large-scale point clouds (δ stands for the maximum octree depth and colors represent height ramp)

Regarding the geometry regularity and the topological relationships, the set $\mathcal{T}_A$

of all architectural symmetries is a subset of the general symmetries $\mathcal{T}$ in Eq. 1:

$$\begin{aligned}
\mathcal{T}_{\mathrm{A}} = \left\{T|\mathcal{A}_{\mathrm{g}}(T) + \mathcal{A}_{\mathrm{t}}(T) < \varepsilon_{\mathrm{A}}, T \in \mathcal{T}\right\} \subseteq \mathcal{T}, \\
\mathcal{A}_{\mathrm{g}}(T) \geq 0, \\
\mathcal{A}_{\mathrm{t}}(T) \geq 0,
\end{aligned} \tag{9}$$

where $\mathcal{T}_{\mathrm{A}}$ is the set of all architectural symmetries, $\mathcal{A}_{\mathrm{g}}$ measures the violations of geometric style, $\mathcal{A}_{\mathrm{t}}$ indicates the violations of topology, and $0 \leq \varepsilon_{\mathrm{A}} \ll 1$ represents a small threshold of error tolerance. The global and local architectural symmetries can be similarly defined. Then the problem of the *global* ASD is:

$$\begin{aligned}
\mathbf{min}\ f(x) &= f_C(x) + \omega\mathcal{A}(x) \\
&= 1 - PCR' + \frac{RMSD'}{diag_C} + \omega\left(\mathcal{A}_{\mathrm{g}}(x) + \mathcal{A}_{\mathrm{t}}(x)\right) \\
\mathbf{s.t.} \quad & x = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^m,
\end{aligned} \tag{10}$$

where the objective function f comprises of symmetry conditions (Eqs. 7 and 8 as f_C) and architectural style conditions (geometric style and topology in Eq. 9 as $\mathcal{A}$), $\omega \in \mathbb{R}^+ \cup \{0\}$ is the relative weight of $\mathcal{A}$. In the formulated global ASD (Eq. 10), the detection of various global architectural symmetries (*e.g.*, reflection, translation, rotation, uniform scaling, and their combinations) is a unified problem. The number m of parameters is usually 3 for either the plain axis of a reflection (*e.g.*, of a plane $ax + by + cz + 1 = 0$ or its normal form $[a, b, c]^\intercal$) or the vector of translation, and 4 for the axis (3D line) of a rotation (Szeliski, 2010). In addition, if the axis of a symmetry is known to be vertical, $m = 2$ can thus be guaranteed. The f_C denotes a combination of the two symmetry conditions in Eqs. 7 and 8 for global ASD, because it is discovered in practice that Eqs. 3 and 4 are not in a monotonic relationship. Local ASD is in the same form of global ASD, e.g., for furniture and building elements (Alexander et al., 2011; Mitra et al., 2013), by replacing C with a subset $C' \subset C$. The formulation thus exposes ASD to not only the approach in this paper but also many other methods.

3.2. Problem-solving using ODAS

Optimization-based Detection of Architectural Symmetries (ODAS, available at https://github.com/ffxue/odas) is an automated approach to solving the formulated ASD problem. The framework of ODAS comprises three stages that were implemented as three functional layers (see Fig. 2). ODAS takes advantage of existing algorithmic libraries such as the Point Cloud Library (PCL, version 1.8.1, available at https://github.com/PointCloudLibrary/pcl), Fast Library for Approximate Nearest Neighbors (FLANN, version 1.7, available at https://www.cs.ubc.ca/research/flann/), and other DFO algorithm libraries

(Muja and Lowe, 2014). As shown in Fig. 2, the data layer consists of the storage of the weighted feature points (*i.e.*, C_v) described in Sect. 3.1, the *k*dtree that indexes the whole input cloud (C) for FLANN search, and the pre-defined geometric regularity (*e.g.*, local symmetries in rooftop elements must be perfectly vertical or horizontal (Xue et al., 2019a)) and topology (*e.g.*, a symmetric window array should reside on a vertical plane of wall (Xue et al., 2018)) in architectural styles. In the model layer, ODAS has a module for computing the objective function (f in Eq. 10) based on the data layer, an adaptor module for incorporating the existing DFO algorithms, and a problem-solving controller for mobilizing all the involved functions for automated ASD. The function layer includes some utilization functions such as symmetry/asymmetry segmentation, detection, and visualization.

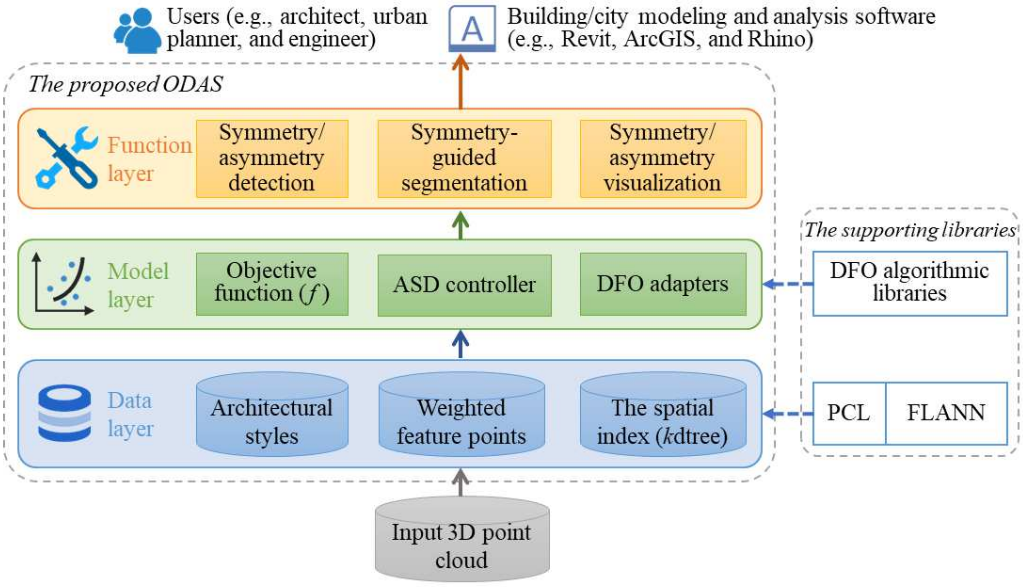

Users (e.g., architect, urban planner, and engineer)
A
Building/city modeling and analysis software (e.g., Revit, ArcGIS, and Rhino)
The proposed ODAS
Function layer
Symmetry/ asymmetry detection
Symmetry-guided segmentation
Symmetry/ asymmetry visualization
The supporting libraries
Model layer
Objective function (f)
ASD controller
DFO adapters
DFO algorithmic libraries
Data layer
Architectural styles
Weighted feature points
The spatial index (kdtree)
PCL
FLANN
Input 3D point cloud

Figure 2: The software architecture of ODAS

ODAS incorporates seven well-known DFO algorithms, as listed in Table 2. The selected DFO algorithms are of three types: (i) global optimization methods such as DIRECT and MLSL-LDS from the library *NLopt*, (ii) population-based metaheuristics including PSO and ABC from the library *Popot*, and (iii) evolution strategies such as NSGA2 and variants of CMAES from the libraries *Nsga2-cpp* and *libcmaes*. The parameters of the algorithms are set to the default values or those in the user manuals. Each DFO algorithm is paired with an adaptor for coding and decoding the symmetry parameters to the variables supported by the algorithmic library. The pairwise parameter voting method in Mitra et al. (2006) is also realized for comparison, while the heuristic feature matching methods are not included due to limitations in application to uncontrolled real-world architectural styles.

Table 2: List of up-to-date algorithms incorporated in ODAS

Algorithm library	Algorithm	Full name	Reference
NLopt (ver. 2.4.2, available at: `https://github.com/stevengj/NLopt/`)	DIRECT MLSL-LDS	DIviding RECTangle Multi-Level Single-Linkage using Low-Discrepancy Sequence	Jones et al. (1993) Kucherenko and Sytsko (2005)
Popot (ver. 2.13, available at: `https://github.com/jeremyfix/popot`)	PSO ABC	Particle Swarm Optimization Artificial Bee Colony	Poli et al. (2007) Karaboga and Basturk (2007)
Nsga2-cpp (ver. 0.2, available at: `https://github.com/dojeda/nsga2-cpp`)	NSGA2	Non-Sorting Genetic Algorithm II	Deb et al. (2002)
libcmaes (ver. 0.9.5, available at: `https://github.com/beniz/libcmaes`)	CMAES sepaIPOP -CMA	Covariance Matrix Adaptation Evolution Strategy A variant of CMAES for noisy problems	Hansen et al. (2003) Hansen (2009)
N.A.	Voting	Pairwise voting-clustering	Mitra et al. (2006)

Alg. 1 shows the pseudocode of the core module, the ASD controller in ODAS, for solving the problem in Eq. 10. The pseudocode describes the high-level functions by omitting the distractive technical details. The four inputs to the ASD controller are: (i) the 3D point cloud C, (ii) the depth of the octree (δ), (iii) the DFO algorithm alg_{DFO}, and (iv) the number of iterations (k) for alg_{DFO}. First, a weighted feature point cloud C_v is sampled using ODAS. Then, the procedure asks alg_{DFO} for promising x (the symmetry parameters) and updates alg_{DFO} with the computed objective value in a k-iteration loop, as shown in Lines 4 to 12 in Alg. 1. The symmetry with the minimal objective value is recorded as the detected symmetry. Because the DFO algorithm alg_{DFO} evolves the meta-model along the ASD procedure, the symmetry parameters converge to the optimum and the symmetry of the point cloud can be detected. Since the time complexity of computing the objective value in Eq. 10 (also see Line 6 in Alg. 1) is $O(4^{\delta}\log n)$, the computational time of the problem-solving procedure of ODAS is bounded at $O(k4^{\delta}\log n)$, which is significantly more efficient than pairwise voting-clustering's $O(n^2)$.

4. Experiments

4.1. Experimental design

Nine cases of real architecture, as listed in Table 3, were selected for validating the efficiency, accuracy, and robustness of ODAS. These cases represent three architectural categories: heritage building, modern building, and infrastructure. The point clouds of the nine cases were retrieved from airborne LiDAR datasets collected by various government agencies or the research team, without any preprocessing such as denoising. The number of points in the cases varies from about 0.01 million to 1.4 million, representing small-scale to large-scale clouds. The densities of the selected point clouds also cover a broad spectrum from 4 points/m^2 to over 2,000 points/m^2. Three cases were LiDAR data on the Hong

Algorithm 1: The ASD controller in ODAS

```
Input  : $C$, $\delta$, $alg_{\mathrm{DFO}}$, and $k$
Output: The global symmetry of $C$
$C_v \leftarrow$ get_octree_sampling ($C$, $\delta$);
$f_{\mathrm{best}} \leftarrow +\infty$;
$x_{\mathrm{best}} \leftarrow \varnothing$;
for $i \leftarrow 1$ to $k$ do                          // $k$ iterations
    $x \leftarrow alg_{\mathrm{DFO}}$.ask_for_a_trial ();
    $f \leftarrow$ compute_f ($x$);                      // using Eq. 10
    $alg_{\mathrm{DFO}}$.tell_and_update ($x$, $f$);
    if $f < f_{\mathrm{best}}$ then                     // a better obj value found
        $f_{\mathrm{best}} \leftarrow f$;
        $x_{\mathrm{best}} \leftarrow x$;                // recording $x$
    end
end
Return $x_{\mathrm{best}}$;
```

Kong 1980 Grid (EPSG:2326) from CEDD (2015), five cases were LiDAR data on the Irish Grid (EPSG:29903) (Laefer et al., 2017), and also one case was a photo-based point cloud (Xue et al., 2019a). The task was set to detect the global reflection symmetries for each case. For quantifying the symmetries and benchmarking the efficiency of the algorithms, two performance metrics were used. One metric is the PCR, *i.e.*, Eq. 3, where the error tolerance $\varepsilon = 0.005$. The other metric is the computational time in single threading mode on a desktop computer with XEON E5-2690 v4 2.6 GHz CPUs, 64 GB memory, Ubuntu 16.04, where the depth of octree sampling was $\delta = 4$. To sum up, this selection of various cases and metrics aims to validate the efficiency, accuracy, and robustness of the proposed ODAS using different ASD scenarios characterized by types of architecture, scales of point cloud, and point densities. The algorithms tested in the experiments included all eight algorithms listed in Table 2.

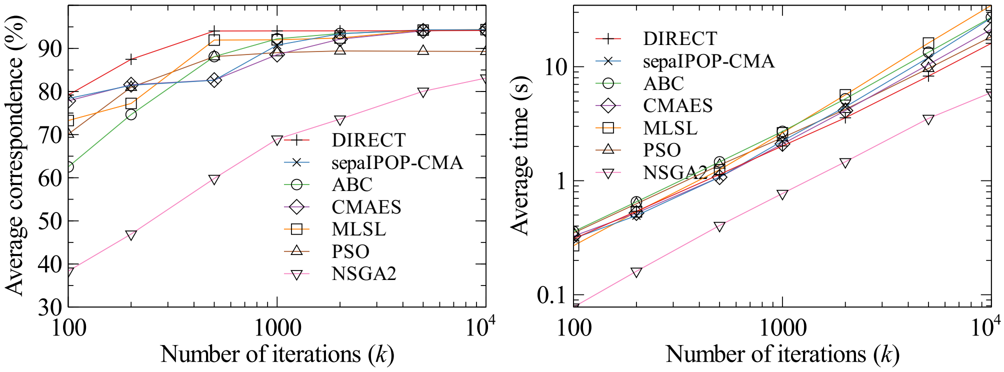
4.2. Benchmarking and comparison of various algorithms

For each test case listed in Table 3, a DFO algorithm was tested with 7 different values of k, ranging from $k = 10^2$ to 10^4, in an exponential order; Using an exponential, rather than a linear or polynomial, variation for benchmarking can expose the ODAS to more extreme scenarios. For each value of k, the resulting metrics, *i.e.*, PCR (Eq. 3) and time, were the average values from 100 independent runs of Alg. 1. The average results of the seven DFO algorithms are shown in Fig. 3. It can be observed that DIRECT is the algorithm found to have the best average correspondence when $k < 2,000$, while sepaIPOP-CMA, ABC, CMAES,

Table 3: List of point clouds of nine cases of symmetric architecture

Category	Id	Name of architecture	n	Density	Geolocation	Type	Spatial ref.	Source
Heritage building	1	Main Building, University of Hong Kong	29,756	4	22.28418, 114.13780	LiDAR	EPSG:2326	CEDD (2015)
	2	Dublin City Hall	459,386	> 100	53.34386, −6.26716	LiDAR	EPSG:29903	Laefer et al. (2017)
	3	Hung Hing Ying Building, University of Hong Kong	1,413,211	$> 2,000$	22.28462, 114.13785	Photo-based		Xue et al. (2019a)
Modern building	4	One George's Quay Plaza, Dublin	1,170,122	> 100	53.34676, −6.25337	LiDAR	EPSG:29903	Laefer et al. (2017)
	5	47-51 O'Connell St. Upper, Dublin	395,818	> 100	53.35135, −6.26167	LiDAR	EPSG:29903	Laefer et al. (2017)
	6	Western District Fruits Wholesale Market, Hong Kong	44,699	4	22.28896, 114.13576	LiDAR	EPSG:2326	CEDD (2015)
Infra-structure	7	Samuel Beckett Bridge, Dublin	570,338	> 100	53.34695, −6.24132	LiDAR	EPSG:29903	Laefer et al. (2017)
	8	Seán O'Casey Bridge, Dublin	223,213	> 100	53.34748, −6.24799	LiDAR	EPSG:29903	Laefer et al. (2017)
	9	Two piers at Victoria Harbor, Hong Kong	12,631	4	22.28963, 114.13557	LiDAR	EPSG:2326	CEDD (2015)

and MLSL-LDS converged to the DIRECT when $k \geq 5,000$. NSGA2 is shown to have fallen behind the other six DFO algorithms in Fig. 3a. Fig. 3b shows the average computational time spent by each DFO algorithm with different k. Log-linear trends can be observed for all the test DFO algorithms, which validates the theoretical time complexity of the proposed method in Sect. 3.1, *i.e.*, $O(k4^{\delta} \log n)$.



(a) Average PCR (higher is better)

(b) Average time (lower is better)

Figure 3: Comparison of the average results of the seven DFO algorithms ($\delta = 4$, 100 independent runs)

Fig. 4 shows the comparison between the proposed approach ODAS (using DIRECT algorithm) and the conventional voting-clustering method. For ODAS, the results of ASD were unsuccessful (below 90% average correspondence) until $k \geq 500$, while the results of voting method were unsuccessful until $\delta \geq 8$. In addition, the average ODAS computational time was hundreds of times shorter than that of the voting methods. Thus, the voting methods were dominated, *i.e.*, inferior in both correspondence and time, by the Pareto frontier formed by ODAS using the DIRECT algorithm. Nevertheless, the voting methods showed efficiency — though unsuccessful — in the bottom left of Fig. 4.

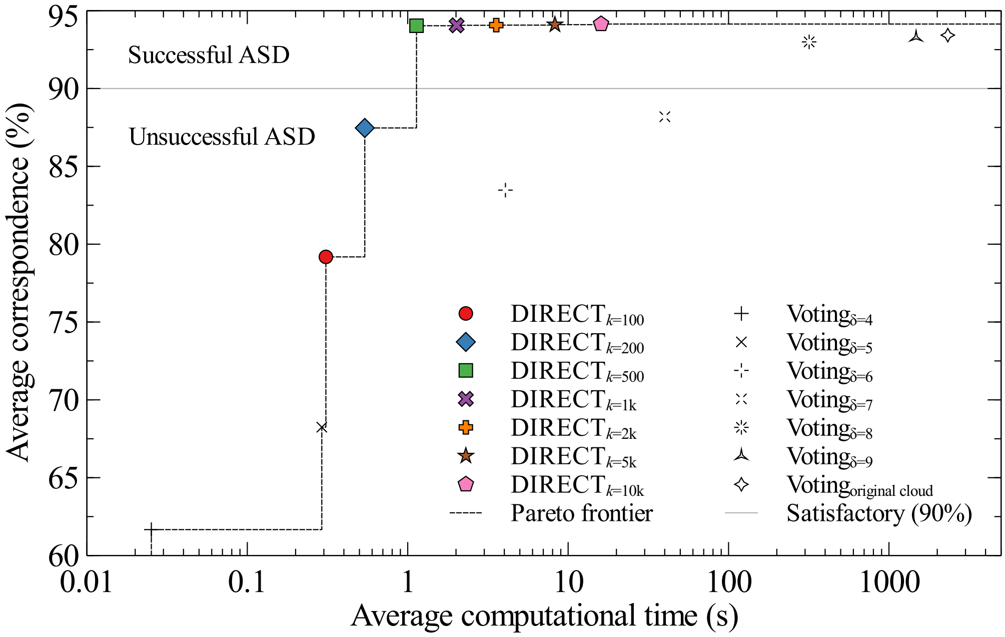


Figure 4: Comparison of average performance metrics between the ODAS using the DIRECT algorithm (δ = 4) and the conventional pairwise voting-clustering method

Table 4 presents a detailed comparison of different methods in the case of the Hung Hing Ying Building, which contains over 1.4 million points. There were two reasons for selecting this case: (i) it was the case with the most (and the most dense) points, and (ii) it was an outlier case on which the ODAS returned a 0.02% lower PCR than the conventional voting method — unlike the average results show in Fig. 4. Because the computational complexity is $O(k4^{\delta} \log n)$, the approach was more than 1,000 times faster than the conventional voting method. It should be noted that the results of ODAS in Table 4 were median values, while the best PCR in the 100 runs was 96.06%. In contrast to the CMAES in Xue et al. (2019a), the approach in this paper improved considerably all the metrics including correspondence, computational time, and RMSD. The reasons for this improvement include an effective octree-based weighted sampling (see Eq. 7), the weighted sum of two symmetry metrics (see Eq. 10), and the very competitive DIRECT algorithm when k is small (see Fig. 3a). In addition, the data on the correspondence and RMSD in Table 4 show that Eqs. 3 and 4 are not monotonic (referring to Sect. 3.1); *e.g.*, Voting$_{\text{original\_cloud}}$ returned a better (higher) correspondence but a worse (higher) RMSD than ODAS.

In summary, the proposed ODAS approach was found to be successful in all

Table 4: Comparison with ASD methods in the literature on the Hung Hing Ying Building case (the best value in each column is in bold)

Method	PCR (%)	Time (s)	RMSD (m)
Voting$_{\text{original\_cloud}}$ (Mitra et al., 2006)	**96.01**	4098.3	0.098
CMAES ($k = 200$) (Xue et al., 2019a)	92.65^a	98.7	0.293
ODAS ($\delta = 4, k = 1,000$, *median* of 100 runs, DIRECT algorithm)	95.99	**3.7**	**0.095**

a: Recalculated with the setting ($\varepsilon = 0.005$) in this paper

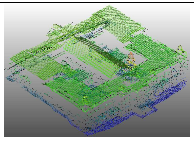
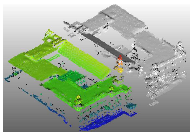
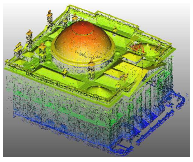
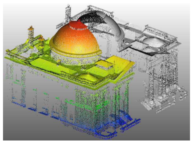
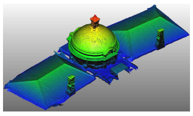
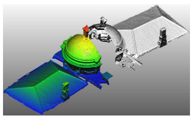
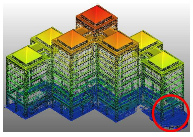
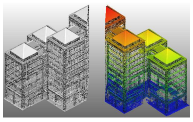
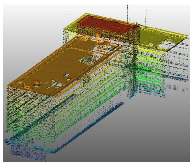
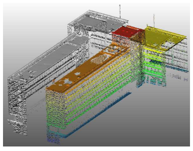
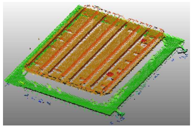
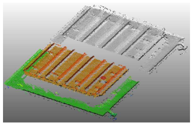
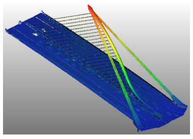
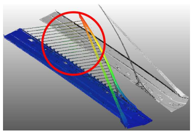
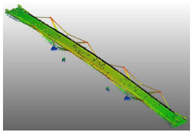
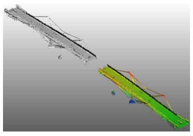
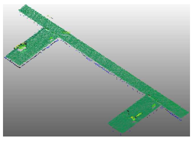
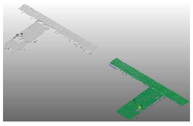
architectural types using modern DFO algorithms including DIRECT, CMAES and its variants, ABC, and MLSL-LDS. DIRECT is the best in terms of accuracy when k is small (*e.g.*, $k < 2,000$), as shown in Fig. 3a, while five algorithms including DIRECT, sepaIPOP-CMA, ABC, CMAES, and MLSL-LDS have very close accuracy when $k \geq 5,000$. Driven by these advanced algorithms, the proposed ODAS is significantly faster and more accurate than conventional voting-clustering methods, as compared in Table 4.

4.3. Results and parameter sensitivity using the DIRECT algorithm

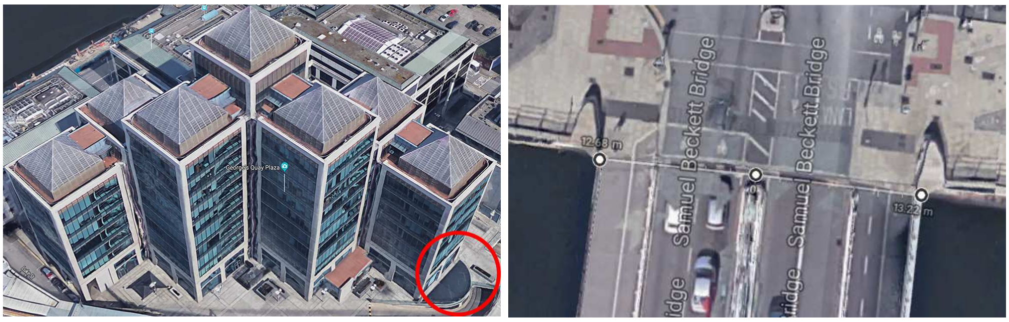
Table 5 shows the median results of global ASD by ODAS using the DIRECT algorithm. The segmented symmetric parts validated that all the detected symmetries were correct and accurate. Furthermore, the overall PCR for the dataset was about 94% in about 2s on average. Also interesting were some asymmetric parts in the results. For example, the point cloud of One George's Quay Plaza (No. 4) lost its as-designed asymmetry, *i.e.*, the carpark entrance, see Table 5 and Fig. 5a, as intended. In the case of Samuel Beckett Bridge (No. 7), all the stay cables surprisingly fell aside the plane axis of the detected global symmetry. An as-built asymmetry, about 0.5 m difference on the bridge deck, was confirmed by an approximate measurement on an aerial photo; see Table 5 and Fig. 5a.

ODAS has two primary parameters, *i.e.*, the depth of octree (δ) and the number of iterations (k) if the DFO algorithm is selected. Fig. 6 shows the variation of the average correspondence and the average computational time under different parameter combinations. Fig. 6a reveals that there exists a satisfactory, steady plateau where $\delta \geq 4$ and $k \geq 1,000$ (in red color). The results of ASD were insensitive to the two parameters if they were on the plateau. Fig. 6b shows the variation of the expected computational time, where the computational time increased with both k and δ. The dashed box in Fig. 6b represents the area of the preferred plateau in Fig. 6a. Fig. 6b shows that some preferred parameter combinations, such as ($\delta = 4, k = 1,000$), spent only a few seconds (between the curves of 1.07 and 3.03 seconds) on average.

Table 5: List of results of global ASD by ODAS (Median of 100 runs)

Id	Thumbnail of input point cloud[a] (C)	Normal[b] of Sym.	Symmetric parts[c] segmented	PCR (%)	Time (s)	Intrinsic[d] asymmetry
1		$\begin{bmatrix}1.026\\-0.046\\0\end{bmatrix}$		86.29	0.81	
2		$\begin{bmatrix}0.171\\0.036\\0\end{bmatrix}$		85.22	1.79	
3		$\begin{bmatrix}-0.080\\0.004\\0\end{bmatrix}$		95.99	3.68	
4		$\begin{bmatrix}-1.979\\2.242\\0\end{bmatrix}$		95.44	2.77	As circled
5		$\begin{bmatrix}0.956\\-3.256\\0\end{bmatrix}$		97.11	2.21	
6		$\begin{bmatrix}0.251\\0.068\\0\end{bmatrix}$		96.96	0.60	
7		$\begin{bmatrix}2.687\\-0.243\\0\end{bmatrix}$		97.51	3.05	As circled
8		$\begin{bmatrix}-0.011\\0.001\\0\end{bmatrix}$		99.49	5.25	
9		$\begin{bmatrix}-6.791\\-1.866\\0\end{bmatrix}$		94.56	0.32	
			Average:	94.29	2.22	

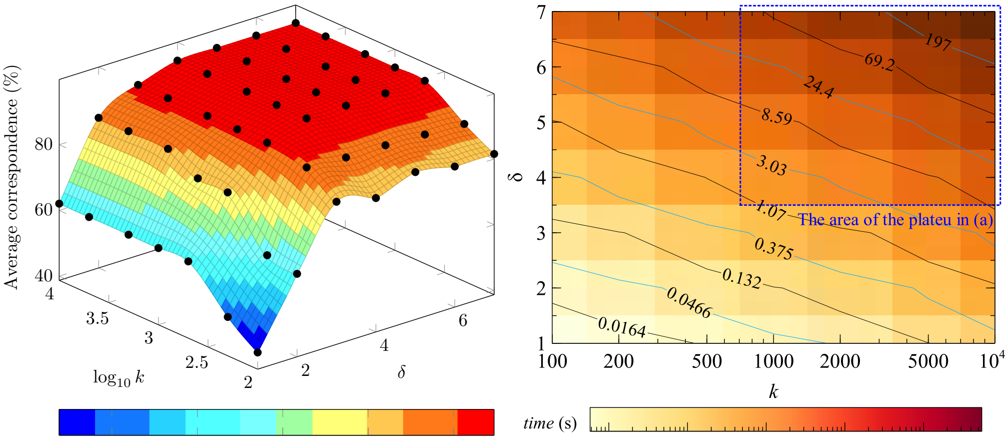
a: Color represents height ramp;
b: Origin is the center, *e.g.*, the symmetry for #1 was $1.026x - 0.046y + 1 = 0$.
c: The front part is in color for comparison;
d: Occlusion and environment were excluded.



(a) As-designed asymmetry (car park entrance) of the No. 4 building

(b) As-built asymmetry on the deck of No. 7, where the east half is about 0.5m wider

Figure 5: Aerial photos about the detected intrinsic, local asymmetries (Source: Google Maps)



(a) Average PCR (higher is better)

(b) Average time (lower is better)

Figure 6: Parameter analysis of the depth of octree (δ) and the number of iterations (k) in ODAS

To differentiate the effects of the two primary parameters of ODAS, Table 6 lists the Pearson's correlations between the parameters and the performance metrics. The δ was compared to logarithms of iterations ($\log_{10} k$) and average time, because a linear variation in δ caused exponential change in the *time*, *i.e.*, $O(k4^{\delta} \log n)$, which was equivalently led by exponential variations in k. The results in Table 6 show that the average PCR had a significant strong positive correlation with the depth of octree (δ). Meanwhile, the logarithm of the average time had a significant strong positive correlation with the depth of octree (δ) and a significant weak positive correlation with the logarithm of iterations ($\log_{10} k$). The correlation between the average PCR and the logarithm of iterations ($\log_{10} k$) was not significant at the 0.01 level. In other words, depth of octree (δ) plays a more critical role than the number of iterations (k) in effective ASD.

Table 6: Pearson's correlations between the two primary parameters of ODAS and the performance metrics ($N = 49$)

Parameter	Statistic	Average PCR (%)	$\log_{10}$ average time/s
δ	Pearson Cor.	0.720^{a}	0.897^{a}
	Sig. (2-tailed)	0.000	0.000
$\log_{10} k$	Pearson Cor.	0.331	0.441^{a}
	Sig. (2-tailed)	0.020	0.001

a: Correlation is significant at the 0.01 level (2-tailed).

5. Discussion

Effective detection of the symmetries omnipresent in architecture and their digital reconstruction in building and city information models (*i.e.*, BIMs and CIMs) has many valuable applications. Currently, available algorithms for ASD from point clouds elicit unsatisfactory accuracy and efficiency. Identification of effective and efficient mathematical algorithms for the problem of ASD will facilitate the development of digital reconstruction for buildings and may improve responses to currently available point cloud processing methods and algorithms.

This paper proposes a novel ODAS approach for ASD with a particular focus on DFO algorithms which showed promising results in a previous study (Xue et al., 2019a). This study extends the mathematical formulation of general symmetry to all types of architectural symmetries by including the conditions about PCR, RMSD, and architectural regularity; in contrast, Mitra et al. (2006) only emphasized on an implicit objective of PCR and the objective in Xue et al. (2019a) was a weighted sum of a variant of RMSD and architectural regularity. Through a combination of octree-based weighted sampling (See Sect. 3.1) and a k-iterated searching strategy (Alg. 1), we bounded the computational time complexity of ODAS to $O(k4^{\delta}\log n)$, which is efficient for processing large-scale point clouds. In contrast, the computational time complexity of Mitra et al. (2006) was $O(n^2)$ and that of Xue et al. (2019a) was $O(k\ \ n\log n)$. The experimental results of a series of tests on three categories of architectures confirmed both the efficiency, *e.g.*, over 1,000 times faster than the voting-clustering (Mitra et al., 2006), and accuracy of ODAS, see Fig. 4 and Table 4.

ODAS includes a long list of state-of-the-art algorithms and will be open source for non-profit applications involving ASD. Users can utilize ODAS freely for the ASD functions with a few lines of code. It offers a broad spectrum of algorithm candidates for different real-life application scenarios based on the parameter sensitivity and correlational analysis shown in Fig. 6 and Table 6. For example, if allowed computation time is tight, *i.e.*, k is small, a user can choose the DIRECT algorithm; if a large k is allowed, other algorithms such as CMAES, ABC, MLSL-LDS, or even a ‘tournament’ (sequential running) of all, can be used.

Besides the approach itself, there were two minor findings. One was the non-monotony between the two well-known symmetry conditions, *i.e.*, Eqs. 3 and 4, in the point clouds of architectures. The other was that that the detected architectural symmetries were also useful to identify the intrinsic asymmetries, such as the as-designed asymmetry of car-park entrance and the as-built asymmetry of bridge deck shown in Table 5 and Fig. 5.

Nevertheless, the proposed ODAS has some limitations that should be considered in future studies:

i. So far, ODAS only works with the classic symmetries with linear or planar symmetry axes. Many intrinsic architectural symmetries are non-classic, *e.g.*, those with L- or J-shape curved symmetry axes or those that are rotational over the storey. Such symmetries are not yet accurately detectable with the pilot version of ODAS. Given that they are increasingly seen in modern architecture, ODAS should be extended in the future to work with non-classic symmetries.

ii. Although the proposed optimization-based approach is competitive in finding one symmetry at a time, it is handicapped in the task of finding repetitions — which can be seen as symmetry compositions of translations and rotations. For example, there can be hundreds of repeated windows and balconies on a modern high-rise building facade. Most mathematical optimization methods, such as the seven included in ODAS, return only one optimal solution, *e.g.*, a translation or a pair of instances, to ASD. However, there exists a set of multi-modal optimization (MMO) algorithms such as variants of PSO (Li and Yao, 2012) that can find all optimal solutions for all repetitions. Therefore, extension of ODAS using MMO algorithms is another future research direction.

iii. The two primary parameters, *i.e.*, k and δ, were tuned to be 1,000 and 4, respectively, based on the experiments in which the error tolerance ε was a constant value of 0.005. However, to what extent the constants can effectively and efficiently achieve ASD for the untested architectural styles remain unclear. For example, the settings might fail if the input point cloud is too small, *e.g.*, $n < 200$. How to adapt the parameters and the approximated symmetry conditions in Eqs. 7 and 8 automatically is another issue.

iv. Last but not least, some DFO algorithms such as CMAES and PSO are known to be unstable, *i.e.*, the results of ASD can be slightly affected by the initial ‘seed’ of the pseudo-random number generator (RNG). Other algorithms such as DIRECT and voting-clustering are more stable. Therefore, the automatic selection of algorithms from ODAS should consider the demands of stability in different application scenarios.

6. Conclusion

This paper introduces a derivative-free optimization (DFO)-based approach for architectural symmetry detection (ASD) from 3D point clouds of real-world architecture. It does so by firstly transforming ASD into a nonlinear optimization problem, and then developing an in-house ODAS (Optimization-based Detection of Architectural Symmetries) approach to solve the formulated problem. The computational experiments on various real-life architecture cases prove that ODAS is significantly faster and more accurate than existing methods, taking only 3.7 seconds to detect global architectural symmetry from a large-scale point cloud of, for example, about 1.4 million points.

The contributions of this paper are three-fold. Firstly, formulating ASD as a nonlinear optimization problem so that numerous state-of-the-art DFO algorithms can be applied constitutes a methodological innovation. Secondly, the up-to-date DFO algorithms developed in this study will be embedded in an open access source so that other researchers can benchmark their work in this field. Thirdly, the ODAS approach can be directly used by software vendors, researchers, and others to develop BIMs and CIMs. By triangulating the ODAS and inexpensive 3D point cloud data, the symmetries and inferred structural and topological semantics may contribute to geometrically accurate, effectively compressed by component abstraction, and semantically rich BIMs and CIMs for various value-added applications. Further studies can be conducted in directions including intrinsic non-classic symmetry detection, multi-modal local symmetry detection, automatic parameter determination, and automatic algorithm selection.

Acknowledgements

This work was supported by the Hong Kong Research Grant Council [grant numbers 17200218 and 17201717] and in part by the University of Hong Kong [grant numbers 201702159013 and 201711159016].

References

Berner Alexander, Wand Michael, Mitra Niloy J., Mewes Daniel, and Seidel Hans-Peter. Shape analysis with subspace symmetries. *Computer Graphics Forum*, 30(2):277–286, 2011. doi: 10.1111/j.1467-8659.2011.01859.x.

Mikhail J. Atallah. On symmetry detection. *IEEE Transactions on Computers*, C-34(7):663–666, 1985. doi: 10.1109/TC.1985.1676605.

Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, 2018. doi: 10.1007/s10514-017-9615-3.

A. Berner, M. Bokeloh, M. Wand, A. Schilling, and H.-P. Seidel. A graph-based approach to symmetry detection. In *Proceedings of the Fifth Eurographics / IEEE VGTC Conference on Point-Based Graphics*, SPBG'08, pages 1–8, Aire-la-Ville, Switzerland, 2008. Eurographics Association. ISBN 978-3-905674-12-5. doi: 10.2312/VG/VG-PBG08/001-008.

Joern Birkmann, Torsten Welle, William Solecki, Shuaib Lwasa, and Matthias Garschagen. Boost resilience of small and mid-sized cities. *Nature*, 537 (7622):605, 2016. doi: 10.1038/537605a.

Martin Bokeloh, Alexander Berner, Michael Wand, H-P Seidel, and Andreas Schilling. Symmetry detection using feature lines. *Computer Graphics Forum*, 28(2):697–706, 2009. doi: 10.1111/j.1467-8659.2009.01410.x.

Christopher M Brown. Inherent bias and noise in the Hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(5):493–505, 1983. doi: 10.1109/TPAMI.1983.4767428.

CEDD. The CEDD 2010 LiDAR survey. Civil Engineering and Development Department, Hong Kong: private communication, 2015.

Ke Chen, Weisheng Lu, Fan Xue, Pingbo Tang, and Ling Hin Li. Automatic building information model reconstruction in high-density urban areas: Augmenting multi-source data with architectural knowledge. *Automation in Construction*, 93:22–34, 2018. doi: 10.1016/j.autcon.2018.05.009.

Alain Chiaradia. Spatial design economies. *Architectural Design*, 79(4):80–85, 2009. doi: 10.1002/ad.921.

Marcelo Cicconet, Vighnesh Birodkar, Mads Lund, Michael Werman, and Davi Geiger. A convolutional approach to reflection symmetry. *Pattern Recognition Letters*, 95:44–50, 2017. doi: 10.1016/j.patrec.2017.03.022.

Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

Mark De Berg, Otfried Cheong, Marc Van Kreveld, and Mark Overmars. *Computational geometry: Introduction*, chapter 1, pages 1–17. Springer, 2008.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

Chuck Eastman, Paul Teicholz, Rafael Sacks, and Kathleen Liston. *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons, 2nd edition, 2011.

Aleksandrs Ecins, Cornelia Fermüller, and Yiannis Aloimonos. Detecting reflectional symmetries in 3d data through symmetrical fitting. In *Proceedings of ICCV Workshop on Detecting Symmetry in the Wild*, pages 1779–1783. IEEE, 2017. doi: 10.1109/ICCVW.2017.210.

Jan Elseberg, Dorit Borrmann, and Andreas Nüchter. One billion points in the cloud–an octree for efficient processing of 3d laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76:76–88, 2013. doi: 10.1016/j.isprsjprs.2012.10.004.

António Ferraz, Clément Mallet, and Nesrine Chehata. Large-scale road detection in forested mountainous areas using airborne topographic LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 112:23–36, 2016. doi: 10.1016/j.isprsjprs.2015.12.002.

Banister Fletcher and Banister F. Fletcher. *A history of Architecture on the comparative method*. Bradbury, Agnew, & Co., London, UK, 5th edition, 1905.

David A Forsyth and Jean Ponce. *Computer vision: A modern approach*. Pearson Education, 2nd edition, 2012.

Nikolaus Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pages 2389–2396. ACM, 2009. doi: 10.1145/1570256.1570333.

Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint*, page arXiv: 1604.00772, 2016.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. doi: 10.1162/106365601750190398.

Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003. doi: 10.1162/106365603321828970.

Jan-Henrik Haunert. A symmetry detector for map generalization and urban-space analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 74:66–77, 2012. doi: 10.1016/j.isprsjprs.2012.08.004.

Xianjin He, Xinchang Zhang, and Qinchuan Xin. Recognition of building group patterns in topographic maps based on graph partitioning and random forest. *ISPRS Journal of Photogrammetry and Remote Sensing*, 136:26–40, 2018. doi: 10.1016/j.isprsjprs.2017.12.001.

Paul VC Hough. Machine analysis of bubble chamber pictures. In *Proceedings of International Conference on High Energy Accelerators and Instrumentation (HEACC 1959)*, pages 554–558. CERN, 1959.

Gordon D James. *The representation theory of the symmetric groups*, volume 682 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin and New York, 1978.

Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of optimization Theory and Applications*, 79(1):157–181, 1993. doi: 10.1007/BF00941892.

Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39(3):459–471, 2007. doi: 10.1007/s10898-007-9149-x.

Michael Kazhdan. An approximate and efficient method for optimal rotation alignment of 3D models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1221–1229, 2007. doi: 10.1109/TPAMI.2007.1032.

Jens Kerber, Michael Wand, Martin Bokeloh, and Hans-Peter Seidel. Symmetry detection in large scale city scans. Technical Report MPI-I-2012-4-001, Max Planck Society for the Advancement of Science, Munich, 2012.

Sergei Kucherenko and Yury Sytsko. Application of deterministic low-discrepancy sequences in global optimization. *Computational Optimization and Applications*, 30(3):297–318, 2005. doi: 10.1007/s10589-005-4615-1.

Debra F Laefer, Saleh Abuwarda, Anh-Vu Vo, Linh Truong-Hong, and Hamid Gharibi. 2015 aerial laser and photogrammetry survey of dublin city collection record. Technical report, New York University, New York, 2017.

Dong-Seop Lee, Luis F Gonzalez, Jacques Periaux, and K Srinivas. Robust design optimisation using multi-objective evolutionary algorithms. *Computers & Fluids*, 37(5):565–583, 2008. doi: 10.1016/j.compfluid.2007.07.011.

Xiaodong Li and Xin Yao. Cooperatively coevolving particle swarms for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 16(2): 210–224, 2012. doi: 10.1109/TEVC.2011.2112662.

Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. Globfit: Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics*, 30(4):52, 2011. doi: 10.1145/2010324.1964947.

Yangbin Lin, Cheng Wang, Jun Cheng, Bili Chen, Fukai Jia, Zhonggui Chen, and Jonathan Li. Line segment extraction for large scale unorganized point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 102:172–183, 2015. doi: 10.1016/j.isprsjprs.2014.12.027.

Yaron Lipman, Xiaobai Chen, Ingrid Daubechies, and Thomas Funkhouser. Symmetry factored embedding and distance. *ACM Transactions on Graphics*, 29 (4):103, 2010. doi: 10.1145/1778765.1778840.

David G Lowe. The viewpoint consistency constraint. *International Journal of Computer Vision*, 1(1):57–72, 1987. doi: 10.1007/BF00128526.

Clément Mallet and Frédéric Bretar. Full-waveform topographic lidar: State-of-the-art. *ISPRS Journal of photogrammetry and remote sensing*, 64(1):1–16, 2009. doi: 10.1016/j.isprsjprs.2008.09.007.

Donald Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129–147, 1982. doi: 10.1016/0146-664X(82)90104-6.

Marc Z Miskin and Heinrich M Jaeger. Adapting granular materials through artificial evolution. *Nature materials*, 12(4):326, 2013. doi: 10.1038/nmat3543.

Niloy J Mitra and Mark Pauly. Symmetry for architectural design. In *Advances in Architectural Geometry*, pages 13–16, Vienna, 2008. TU Wien.

Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics*, 25(3):560–568, 2006. doi: 10.1145/1141911.1141924.

Niloy J Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. Symmetry in 3D geometry: Extraction and applications. *Computer Graphics Forum*, 32(6):1–23, 2013. doi: 10.1111/cgf.12010.

Aron Monszpart, Nicolas Mellado, Gabriel J Brostow, and Niloy J Mitra. RAPter: rebuilding man-made scenes with regular arrangements of planes. *ACM Transactions on Graphics*, 34(4):103, 2015. doi: 10.1145/2766995.

Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014. doi: 10.1109/TPAMI.2014.2321376.

SV Nghiem, SH Yueh, R Kwok, and FK Li. Symmetry properties in polarimetric remote sensing. *Radio Science*, 27(05):693–711, 1992. doi: 10.1029/92RS01230.

Giuseppe Nicosia and Giovanni Stracquadanio. Generalized pattern search algorithm for peptide structure prediction. *Biophysical Journal*, 95(10):4988–4999, 2008. doi: 10.1529/biophysj.107.124016.

Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm Intelligence*, 1(1):33–57, 2007. doi: 10.1007/s11721-007-0002-0.

Dan Raviv, Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Full and partial symmetries of non-rigid shapes. *International Journal of Computer Vision*, 89(1):18–39, 2010. doi: 10.1007/s11263-010-0320-3.

Luis Miguel Rios and Nikolaos V Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013. doi: 10.1007/s10898-012-9951-y.

Aparajithan Sampath and Jie Shan. Building boundary tracing and regularization from airborne LiDAR point clouds. *Photogrammetric Engineering & Remote Sensing*, 73(7):805–812, 2007. doi: 10.14358/PERS.73.7.805.

Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, 2007. doi: 10.1111/j.1467-8659.2007.01016.x.

Zeyun Shi, Pierre Alliez, Mathieu Desbrun, Hujun Bao, and Jin Huang. Symmetry and orbit detection via Lie-algebra voting. *Computer Graphics Forum*, 35(5): 217–227, 2016. doi: 10.1111/cgf.12978.

Philip Steadman. *Architectural morphology: An introduction to the geometry of building plans*. Taylor & Francis, 1983.

Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

Pingbo Tang, Daniel Huber, Burcu Akinci, Robert Lipman, and Alan Lytle. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, 19(7):829–843, 2010. doi: 10.1016/j.autcon.2010.06.007.

I Toschi, MM Ramos, E Nocerino, F Menna, F Remondino, K Moe, D Poli, K Legat, Francesco Fassi, et al. Oblique photogrammetry supporting 3d urban reconstruction of complex scenarios. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII (1/W1):519–526, 2017. doi: 10.5194/isprs-archives-XLII-1-W1-519-2017.

Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011. doi: 10.1111/j.1467-8659.2011.01884.x.

Rebekka Volk, Julian Stengel, and Frank Schultmann. Building Information Modeling (BIM) for existing buildings—literature review and future needs. *Automation in Construction*, 38:109–127, 2014. doi: 10.1016/j.autcon.2013.10.023.

Jun Wang, Yabin Xu, Oussama Remil, Xingyu Xie, Nan Ye, C Yi, and M Wei. Automatic modeling of urban facades from raw LiDAR point data. *Computer Graphics Forum*, 35(7):269–278, 2016. doi: 10.1111/cgf.13024.

Yanzhen Wang, Kai Xu, Jun Li, Hao Zhang, Ariel Shamir, Ligang Liu, Zhiquan Cheng, and Yueshan Xiong. Symmetry hierarchy of man-made objects. *Computer Graphics Forum*, 30(2):287–296, 2011. doi: 10.1111/j.1467-8659.2011.01885.x.

Uwe Weidner and Wolfgang Förstner. Towards automatic building extraction from high-resolution digital elevation models. *ISPRS journal of Photogrammetry and Remote Sensing*, 50(4):38–49, 1995. doi: 10.1016/0924-2716(95)98236-S.

Qiaoyun Wu, Hongbin Yang, Mingqiang Wei, Oussama Remil, Bo Wang, and Jun Wang. Automatic 3D reconstruction of electrical substation scene from LiDAR point cloud. *ISPRS Journal of Photogrammetry and Remote Sensing*, 143:57–71, 2018. doi: 10.1016/j.isprsjprs.2018.04.024.

Y Xu, L Hoegner, S Tuttas, and U Stilla. Voxel- and graph-based point cloud segmentation of 3D scenes using perceptual grouping laws. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, IV (1/W1):43–50, 2017. doi: 10.5194/isprs-annals-IV-1-W1-43-2017.

Yusheng Xu, Wei Yao, Sebastian Tuttas, Ludwig Hoegner, and Uwe Stilla. Unsupervised segmentation of point clouds from buildings using hierarchical clustering based on Gestalt principles. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(11):4270–4286, 2018. doi: 10.1109/JSTARS.2018.2817227.

Fan Xue, Weisheng Lu, and Ke Chen. Automatic generation of semantically rich as-built building information models using 2D images: A derivative-free optimization approach. *Computer-Aided Civil and Infrastructure Engineering*, 33(11):926–942, 2018. doi: 10.1111/mice.12378.

Fan Xue, Ke Chen, and Weisheng Lu. Architectural symmetry detection from 3D urban point clouds: A derivative-free optimization (DFO) approach. In *Advances in Informatics and Computing in Civil and Construction Engineering*, pages 513–519. Springer, 2019a. doi: 10.1007/978-3-030-00220-6_61.

Fan Xue, Weisheng Lu, Ke Chen, and Anna Zetkulic. From 'semantic segmentation' to 'semantic registration': A derivative-free optimization-based approach for automatic generation of semantically rich as-built building information models (BIMs) from 3D point clouds. *Journal of Computing in Civil Engineering*, 2019b. Accepted, in press.

Kazunori Yamaguchi, Tosiyasu L Kunii, Kikuo Fujimura, and Hiroshi Toriya. Octree-related data structures and algorithms. *IEEE Computer Graphics and Applications*, 4(1):53–59, 1984. doi: 10.1109/MCG.1984.275901.

Hao Zhang, Kai Xu, Wei Jiang, Jinjie Lin, Daniel Cohen-Or, and Baoquan Chen. Layered analysis of irregular facades via symmetry maximization. *ACM Transactions on Graphics*, 32(4):121, 2013. doi: 10.1145/2461912.2461923.