

Layered Analysis of Irregular Facades via Symmetry Maximization

Hao Zhang¹ Kai Xu^{2,3*} Wei Jiang³ Jinjie Lin² Daniel Cohen-Or⁴ Baoquan Chen^{2*}
¹Simon Fraser University ²Shenzhen VisuCA Key Lab / SIAT ³HPCL, NUDT ⁴Tel Aviv University

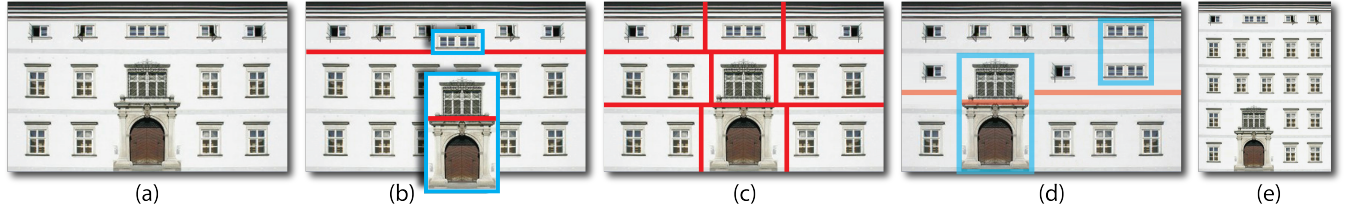


Figure 1: Symmetry-driven structural analysis of an irregular facade (a) results in a hierarchical decomposition (b) into regular grids. Our analysis introduces layering (b), going beyond flat segmentation via splits (c) and allowing more compact and natural structural representations. The resulting hierarchical model of facades enables applications such as structural editing (d) and retargeting (e).

Abstract

We present an algorithm for *hierarchical* and *layered* analysis of *irregular* facades, seeking a high-level understanding of facade structures. By introducing layering into the analysis, we no longer view a facade as a flat structure, but allow it to be structurally separated into depth layers, enabling more compact and natural interpretations of building facades. Computationally, we perform a symmetry-driven search for an optimal hierarchical decomposition defined by split and layering operations applied to an input facade. The objective is *symmetry maximization*, i.e., to maximize the sum of symmetry of the substructures resulting from recursive decomposition. To this end, we propose a novel *integral symmetry* measure, which behaves well at both ends of the symmetry spectrum by accounting for all partial symmetries in a discrete structure. Our analysis results in a structural representation, which can be utilized for structural editing and exploration of building facades.

CR Categories: I.3.8 [Computing Methodologies]: Computer Graphics—Applications;

Keywords: hierarchical structural analysis, irregular facades, layering, symmetry maximization, integral symmetry

Links: [DL](#) [PDF](#) [WEB](#) [VIDEO](#) [DATA](#) [CODE](#)

1 Introduction

High-level processing of shapes or patterns has been receiving increasing interests in computer graphics. At the core of these approaches is an analysis to understand the structure of an input. The understanding leads to an effective reuse of data for structure editing, synthesis, or exploration. An interesting class of structures that

has received much attention lately is that of building facades [Musiakski et al. 2012]. While the fundamental building blocks of facades are regular grids of windows or balconies, real-world facades exhibit an amazing variety of *irregular* mixtures of grid structures. The ubiquity of facades, combined with the rich irregularities therein, makes them useful and intriguing structures to study.

In this paper, we develop an algorithm for analyzing irregular 2D facades. Our goal is to obtain a high-level understanding or explanation of the *structure*, rather than appearance, of a facade. The fundamental analysis task involves grouping or decomposition of the basic structural elements of a facade, e.g., windows and balconies. Our structural decomposition is *hierarchical* and it is built on two fundamental operations: *split* and *layering*; see Figure 1. By introducing layering into the analysis, we no longer view a facade as a flat structure, but allow it to be structurally separated into depth layers. Layering is motivated by cognitive theories of visual completion [Buffart et al. 1983], positing the mind’s intention to complete regular patterns. It enables a more compact and natural interpretation of a frequent pattern in facades, where the regularity of a structure (e.g., a grid of windows) is interrupted by certain elements (e.g., a door), e.g., see Figure 1(b) vs. 1(c).

Our hierarchical analysis algorithm is *symmetry-driven*: we recursively decompose facade elements based on measures of symmetry or repetition. This is inspired by Gestalt Law of Prägnanz, which emphasizes the prevalence of symmetry and regularity in perceptual grouping [Wertheimer 1923]. However, unlike previous works aimed at regularity detection [Pauly et al. 2008; Wu et al. 2010] or flat segmentation of facades into regular grids [Chao et al. 2012], our algorithm focuses on the challenge of analyzing irregularity. Specifically, we seek a high-level explanation of the irregular arrangements and overlays of the grids.

Law of Good Gestalt and simplest explanation. Computationally, we pose the analysis problem as that of finding an optimal hierarchical decomposition of a facade. The optimization has a twofold objective. First, we seek the most perceptual decomposition. Our approach follows the well-known *Law of Good Gestalt* [Wertheimer 1923] which stipulates that one should maximize the simplicity, regularity, or orderliness of the substructures resulting from each decomposition. Our second objective is to seek the best structural explanation which, according to *Occam’s Razor*, is often the *simplest* one. In our work, we equate simplicity to the minimization of the number of decompositions.

*Corresponding authors: kaixu@nudt.edu.cn, bq.chen@siat.ac.cn

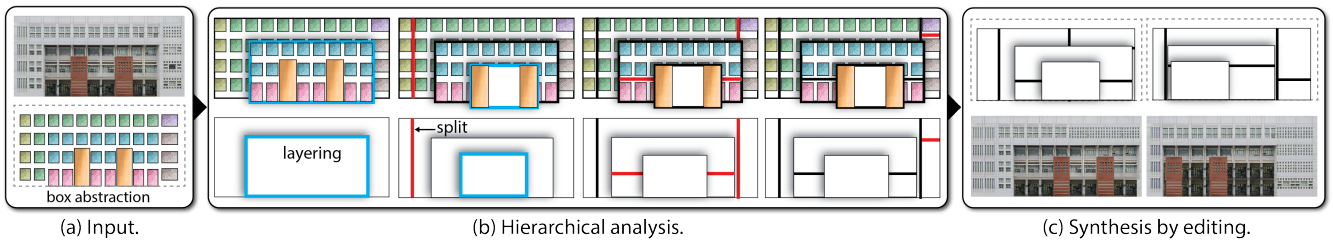


Figure 2: Overview of facade analysis and synthesis. Input to the analysis is a box abstraction, with boxes having the same color representing repeated basic elements of a facade, e.g., a window. Our algorithm computes an optimal hierarchical binary decomposition (b: top) based on splitting (dark line) and layering (shadowy box) of facade structures enclosed by the boxes. The analysis result defines a structural representation (b: bottom), which can be altered (c: top) to produce interesting structural variations (c: bottom).

Symmetry maximization. As we quantify regularity by symmetry, the first objective dictates that we should maximize the symmetry of the substructures. Interestingly, such a symmetry maximization leads to fulfillment of the second objective as well. To explain a facade, the structural decomposition should naturally stop when a pattern requires no further explanation. In our setting, this is a pattern that represents, or is perceived as, a regular grid, which possesses a high degree of symmetry; see Figure 2(b)-top. In the latter case, the pattern is an incomplete grid of identical elements and naturally perceived as a completed grid as posited by the Gestalt *Law of Closure*. To minimize the number of decompositions, or equivalently, to reach regular grids as quickly as possible, we seek symmetry maximization at each decomposition step as well.

Consequently, our optimization problem seeks a hierarchical *binary* decomposition, via layering and splits, which maximizes the sum of symmetry of all the substructures in the obtained hierarchy. The main challenges of the problem are threefold:

- *Symmetry measure:* The optimization requires a symmetry measure to quantify both near-regular and irregular structures. Existing measures [Kazhdan et al. 2004; Podolak et al. 2006; Simari et al. 2006; Graham et al. 2010] have been designed to quantify how close a shape is to possessing a global symmetry. We propose a novel *integral symmetry* measure, which behaves well at both ends of the symmetry spectrum by accounting for all partial symmetries in a discrete structure.
- *Layering:* Existing structural analyses of facades are mainly based on procedural models, notably split grammars [Wonka et al. 2003; Teboul et al. 2011]. Such models only produce hierarchical flat subdivisions of a facade. Layering substantially extends the search space. It also poses the structure completion problem resulting from removing a top layer.
- *Hierarchy optimization:* The optimization turns out to be a computationally intensive problem. To this end, we develop a genetic algorithm where evolution of the solutions is driven by the integral symmetry measure.

Structural representation. The result of our analysis implies a *structural representation*. We view the decompositions as being applied to the *blank* bounding boxes of the facade elements; see Figure 2(b)-bottom vs. 2(b)-top. Specifically, the box which bounds the input facade is recursively transformed into more and more boxes. The representation contains no image content; it defines a structure. Combining the representation with an *instantiation* of the finest-level boxes by image content, we obtain a *generative model* for the input facade — it offers an explanation of how the input facade was seemingly generated; see Figure 2(b).

Contributions. The main contributions of our work are:

- Hierarchical structural analysis of irregular facades via symmetry maximization.
- Introduction of layered analysis, which leads to simpler explanation of facade structures.
- An integral symmetry measure designed to be applicable to both near-regular and irregular discrete structures.
- A structural representation for facade structures, which defines a generative model.

An immediate application of the structural representation is facade variation, which is achieved by altering the split and layering operations, as shown in Figure 2(c). We demonstrate this with an interactive editing tool. We also develop a few other applications which utilize the hierarchical structural representation we obtain, including facade retargeting, retrieval, and exploration. Interactive demonstrations can be found in the accompanying video.

2 Related work

Symmetry analysis has received much attention lately [Mitra et al. 2012]. Most earlier works on structural symmetry detection [Pauly et al. 2008; Wu et al. 2010] extract repeated patterns without organization. It is widely believed that human perception operates in a hierarchical manner [Palmer 1977; Hochstein and Ahissar 2002], motivating hierarchical analysis. Simari et al. [2006] compute a folding mesh hierarchy by recursively computing the dominant reflectional symmetry in a sub-shape. Martinet [2007] constructs a structural hierarchy to discover congruent scene components and obtain a compact scene representation. The symmetry hierarchy of Wang et al. [2011] results from a bottom-up analysis via recursive symmetry grouping and part assembly, which are guided by heuristic rules without an objective function. None of the works above define or utilize a symmetry measure for discrete structures.

Existing works on facade analysis mostly focus on regularity detection [Pauly et al. 2008; Wu et al. 2010] or flat segmentation of facades [Wonka et al. 2003; Teboul et al. 2011; Shen et al. 2011; Chao et al. 2012]. Hierarchical subdivisions of facades can be obtained but only through split operations. Computation of the splits is guided either by rules from split grammars [Wonka et al. 2003; Teboul et al. 2011] or user interaction [Shen et al. 2011]. Our consideration of layering is motivated by concepts from visual completion. Most global theories are modifications of the Gestalt laws of grouping in which symmetry plays an important role [Wertheimer 1923]. In the context of facade analysis, the globally predicted percepts would point to completion that achieves the highest degree of symmetry possible [Buffart et al. 1983]. Besides introducing layering into the analysis, our work focuses on understanding the generally irregular organization of detected regular grids.

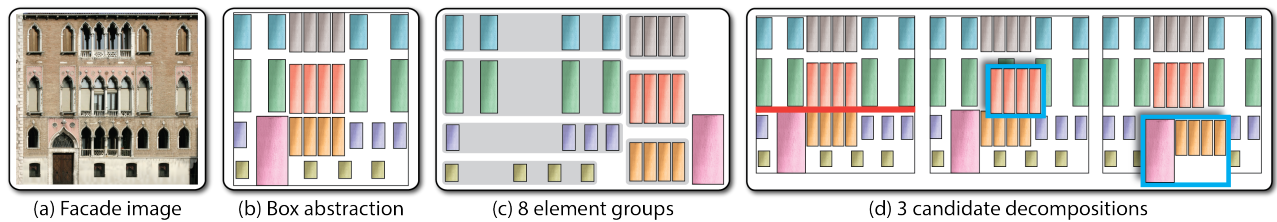


Figure 3: *Box abstraction and decomposition candidates.* A facade image (a) is converted into a box abstraction (b), where the boxes represent the atomic elements. Element groups (c) formed by repeated elements (boxes in same color) in rectangular grids are identified. The four groups on the left are incomplete and those on the right are complete. Decomposition candidates are selected (d) and incomplete element groups undergo structural completion (partially revealed by slightly moving the top layer box to the side).

Recent works on inverse procedural modeling also aim to recover generative models for a shape or structure. Stava et al. [2010] generate a parametric L -system which reproduces a given 2D line art. Bokeloh et al. [2010] perform local similarity and symmetry search to decompose a 3D model and synthesize new shapes procedurally via insertion, deletion, and replacement of structural elements. Similar to the construction of a symmetry hierarchy [Martinet 2007; Wang et al. 2011], their analyses take into account detected symmetries but are not guided by a structural symmetry measure. Also, these analyses do not consider layering which considerably increases the search complexity and at the same time, simplifies the explanation of certain structures. The recovered shape grammar only consists of rewriting rules which re-generates and re-distributes structural elements in a single layer.

Many works on symmetry analysis rely on a symmetry measure. When detecting prominent reflectional symmetries in a shape, one often measures the extent a given shape is reflectionally symmetric with respect to a given axis by the extent of overlap between the shape and its reflection [Kazhdan et al. 2004; Simari et al. 2006; Podolak et al. 2006]. To the best of our knowledge, existing symmetry or asymmetry measures, like the one above, only apply to an individual shape. In contrast, our integral symmetry measure applies to a set of elements and integrates inter- and intra-element symmetries. Moreover, existing continuous symmetry measures [Graham et al. 2010] focus on computing the “effort” it takes to bring a shape into perfect symmetry and as such, they are more suited to shapes that are close to being symmetric.

An increasing number of works modify or generate new man-made structures from exemplars where efforts are invested to preserve the input structures, in particular the symmetries or nearly perfect grid structures therein [Harada et al. 1995; Aliaga et al. 2007; Wu et al. 2010; Bokeloh et al. 2012]. Early work by Harada et al. [1995] explores design spaces specified by a given shape grammar based on a mixed continuous-discrete model. Most recent work of Bokeloh et al. [2012] develops an editing tool which adapts the structure of the input, while dealing with translational repeated patterns only. The work of Lin et al. [2011] demonstrates the importance of hierarchical models in structure-aware retargeting of irregular architecture. However, their hierarchical model was constructed manually. Even layered facade *synthesis* had appeared before, e.g., in Parisk and Müller [2001], via semi-automatic composition of texture layers. Our work allows automated structural analysis of irregular facades of a much richer variety and the analysis results facilitate several applications including retargeting.

3 Structural analysis

In this section, we describe our structural facade analysis algorithm via symmetry maximization. The symmetry measure which defines the objective of the optimization is described in Section 4.

3.1 Overview

Input. With a focus on structural analysis, our algorithm operates on a structural *abstraction* of facade images. We call it a *box abstraction* as it consists of a set of axis-aligned 2D boxes tightly enclosing the basic structural elements of a facade; see, e.g., Figures 2(a) and 3(b). We develop a semi-automatic tool to obtain the initial set of boxes from a facade image; see Section 3.2.

Element groups. An *element group* is formed by a set of well-aligned boxes whose content repeats. Such a group is either a complete or partial regular grid, but it must be *maximal*, i.e., one cannot extend the group using an extra box; see Figure 3(c). In this paper, we only consider element groups conforming to regular *rectangular* grids. Element groups are identified by the interactive tool.

Candidate selection. Even for an input with a moderate number of boxes, the total number of all possible split and layering operations is very large. We reduce the cost of optimization by producing a restricted set of candidate decompositions which form the search space. Candidate selection follows the law of grouping by regularity. Constraints are defined based on the element groups; see Figure 3(c)-(d). More details are provided in Section 3.3.

Hierarchy optimization. Starting from the input box abstraction, we recursively decompose its bounding box together with the set of boxes therein. The box abstraction within each resulting box is referred to as a *substructure*. Our objective is to find a hierarchical binary decomposition which maximizes the sum of symmetry of the substructures in the hierarchy. Even after restricting the search to only the candidate decompositions, the total number of decomposition trees is still very large, making the optimization problem computationally intensive. In our work, we pursue a heuristic solution and resort to *genetic algorithms*. We evolve the solution space by top-down probabilistic generation of decomposition hierarchies, where a sampling bias is introduced by our symmetry measure. The fitness function is defined by the optimization objective. Section 3.4 provides more details on structural decomposition.

3.2 Interactive box abstraction

Given a facade image, we construct an axis-aligned box abstraction. Each box tightly encloses an *atomic* element of the facade, e.g., a window or balcony, which is indivisible throughout the analysis. Restricting the analysis to axis-aligned structures simplifies low-level feature analysis, while still allowing a large variety of facades to be studied. Computing the abstraction requires detection of atomic elements and their repetitions. We had experimented with existing schemes for automatic facade segmentation, e.g., [Teboul et al. 2011; Shen et al. 2011], but found the results on irregular facades to be less than satisfactory. Hence, we have developed an

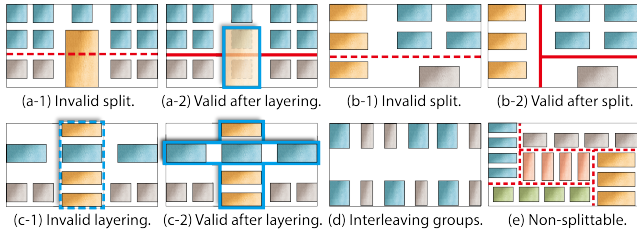


Figure 4: Examples of invalid decompositions becoming valid after other decompositions (a-c). (d)-top: two incomplete element groups partially cover each other. (d)-bottom: two complete groups without any partial coverage. Both cases in (d) show interleaving groups and either group can be a top layer candidate. (e): no valid splits can explain the structure as five grids as shown.

interactive tool for box abstraction.

In iterative fashion, the user draws a tight bounding box over a seed (atomic) element. The box is then slid horizontally to interactively search for similar enclosed content as the seed over the entire input image. Automatic matching relies on normalized pixel-wise square difference to measure box-to-box (content) similarity. To tolerate a higher degree of appearance variation among repeated elements, we allow the user to draw scribbles over these elements to localize the search: the matching window must contain the scribbles. At the end, the user has the option to further adjust the abstraction results. All the boxes that are deemed to be repetitions of each other are shown in the same color, e.g., see Figure 3(b). The accompanying video demonstrates the interactive tool.

3.3 Decomposition candidate selection

Given a box abstraction, we first identify the element groups. If a group represents an incomplete grid, we apply structure completion so that the completed grid is evaluated during symmetry-driven hierarchy optimization. The spatial arrangement of the element groups determines where box splits and layering may occur, resulting in a set of decomposition candidates; see Figure 3(d).

Element groups. With the atomic elements at the lowest level, element groups are the next higher-level structural primitives. In our analysis, repetition detection is executed entirely in the interactive box abstraction step. If an atomic element does not have a sufficiently similar counterpart, it forms an element group by itself. In most cases, a set of repeated boxes detected via box abstraction already form an element group. The user also has the option of specifying new box groupings interactively by drawing virtual boxes on top of the facade image. Of course, automatic scheme for grid detection, e.g., [Pauly et al. 2008], can also be employed.

Structure completion. Given an incomplete element group, i.e., a rectangular partial grid, it is fairly straightforward to fill in the missing elements to form a complete and approximately regular grid. The regularity is only approximate since in general, the empty space for fill-ins does not allow a perfectly regular spacing between all elements in the group. We insert new elements in a way to ensure alignment (if needed) and minimize the discrepancies in the spacings, similarly to schemes for distributing repeated elements in retargeting applications [Wu et al. 2010; Lin et al. 2011].

Coverage between groups and depth order. The incompleteness of an element group G_1 is caused by some elements from another group G_2 interrupting the regularity of G_1 as a complete regular grid. In other words, some element of G_1 is “occluded” by some element of G_2 . In this case, we say that G_2 *partially covers*

G_1 . If G_1 is partially covered by G_2 but not vice versa, then we say that G_2 (completely) *covers* G_1 . In this case, there is a clear *depth order* between the two groups: G_2 is in front of G_1 . Coverage relations induce layering and help us select layering candidates.

Principles of candidate selection. There are two key principles behind our selection of valid decomposition candidates:

P-1: An element group, which is perceived as a whole by Gestalt law of perceptual grouping, cannot be divided into two components by a valid decomposition.

P-2: Candidate selection is carried out *recursively*.

Hence the selection of validity decompositions is confined to the substructure being analyzed. The selection is made among all possible decompositions appropriate for the substructure, following principle P-1. Note that a decomposition may be invalid in the current substructure but becomes valid further down in the decomposition hierarchy, as shown in Figures 4(a)-(b).

Layering candidates. A layering operation splits a flat substructure, along the depth direction, into two overlapping layers. Following principle P-1, given a substructure T , any subset of element groups belonging to T can form a layering candidate for T . However, this results in an exponential number of layering candidates. We reduce the candidate set by examining the coverage relation between element groups. Given two element groups G_1 and G_2 , we consider the following cases:

1. G_1 covers G_2 or G_2 covers G_1 , e.g., Figures 4(a-2) and (c-2).
2. G_1 partially covers G_2 and vice versa;
3. neither element group partially covers the other, but the bounding boxes of G_1 and G_2 intersect.

Clearly, in all other cases, the bounding boxes of G_1 and G_2 do not intersect, thus no layering between them should happen. Case 2 implies that G_1 and G_2 are both incomplete groups; see Figure 4(d)-top. This case is quite rare in real-world facades; in fact, we have not found any real-world examples in our dataset. Case 3 implies that both G_1 and G_2 are complete groups; see Figure 4(d)-bottom. Cases 2 and 3 are both examples of *interleaving* element groups and a coverage relation between the groups cannot be determined — it is a “tie”. When selecting layering candidates, we only consider element groups that either cover or interleave other groups. Any subset of such groups can form a layering candidate.

Split candidates. A split divides the current substructure into two non-empty and non-overlapping substructures. In our analysis, we require each split to go across the horizontal or vertical extent of the bounding box. Principle P-1 above implies that splits should only serve to separate different element groups without dividing any other group. Hence, a split on a substructure T is valid if it does not intersect the bounding box of any element group belonging to T . Figure 3(d) and Figures 4(a)-(c) show a few characteristic examples of valid and invalid decompositions.

3.4 Optimal hierarchical decomposition

Given a box abstraction, we search for the optimal hierarchical binary decomposition via symmetry maximization. Construction of a hierarchy is top-down and the search at each node is restricted to the set of decomposition candidates appropriate for the substructure at that node. The decomposition stops when the substructure needs no further explanation, i.e., it is an element group.

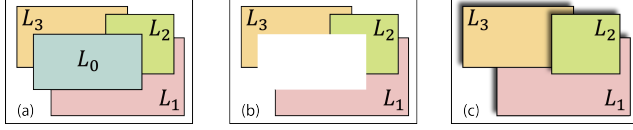


Figure 5: Resolving depth among multiple overlapping element groups when evaluating layering operations. Four groups (abstracted as boxes) with L_0 as the top layer for a decomposition (a). Removing L_0 leaves a gap (b). Structure completion after temporary depth resolution, which is based on a simple heuristic: element groups with smaller areas go in front.

The key measure in the optimization is a *symmetry score* defined for each decomposition; this is defined in Section 4. The *objective function* is the sum of symmetry scores of all interior nodes of a decomposition hierarchy. Hence the optimization problem is:

$$\arg \max_{T: \text{a hierarchy}} \sum_{\mathbf{n}: \text{an interior node of } T} \text{SymScore}(\mathbf{n}). \quad (1)$$

The most straightforward approach to symmetry maximization is greedy optimization, where at each node, we seek the decomposition which maximizes the symmetry score. However, to alleviate the local minima problem, we employ a genetic algorithm.

Genetic algorithm. The genetic algorithm operates on an evolving population composed of decomposition hierarchies (trees). To start, we sample $s = 30$ trees to form the initial population. Stochastic construction of a sample tree is performed top-down. Starting at the root and recursively down the tree, we apply *importance sampling* at each node based on the symmetry scores defined for all the candidate decompositions — higher probabilities are assigned to candidates with higher symmetry scores.

Tree *mutation* occurs at a random node and replaces the current decomposition by another randomly chosen candidate. Once a node is mutated, its subtree is updated via top-down importance sampling. *Crossover* is performed between two trees by exchanging two subtrees rooted at identical substructures. To evaluate the *fitness* of a tree, we use the objective function of our optimization. We execute a steady-state genetic algorithm to evolve the tree population for $g = 25$ generations, where the top 50% most fit trees are migrated from the current generation to the next. The remaining population is filled with newly created trees via mutation and crossover.

Lazy structure completion. While structure completion for an individual element group is straightforward, the problem of how to structurally complete the “occluded” space left after a top layer is removed is generally quite difficult. The difficulty arises only when three or more element groups are overlapping each other. Figure 5(a) shows an example with four such groups, where L_0 represents the top layer for a decomposition \mathbf{n} . To evaluate the symmetry of the bottom layer, we need to fill in the “gap” by resolving the depth order among L_1 , L_2 , and L_3 . One could formulate the problem as finding the order leading to the most symmetric completion. This is however a global optimization problem.

We take a *lazy* approach (as in lazy evaluation) by relying on a simple heuristic: the group whose bounding box has the smaller size is always in front; see Figure 5(c). The resulting depth order leads to a *temporary* completion so that the symmetry score at \mathbf{n} can be computed. Importantly, once a full tree is constructed, the depth order among all groups, including L_1 , L_2 , and L_3 , is resolved based on symmetry maximization. Now we re-compute all the symmetry scores based on the actual depth order and the structural completions implied from it. The genetic algorithm then uses the accurate

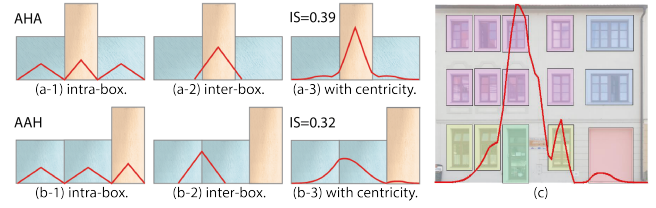


Figure 6: Plots of symmetry profiles. (a-b) Profiles of intra-, inter-box symmetries, and their integration with centrality, for two simple box patterns of the types “AHA” (top) and “AAH” (bottom). (c) A more complex pattern and its symmetry profile.

fitness score for the solution search. This shows another advantage of using genetic algorithm over greedy optimization.

Simplest explanation. Recall that our explanation of a facade structure goes no further than a regular grid. Since each decomposition tree T considered in our optimization terminates at a set of maximal regular grids (they are the element groups), it is not difficult to show that T would terminate the “fastest” in that it requires the minimum number of decompositions. In other words, T always provides the simplest explanation in our terminology. In fact, all the trees considered in our search contain the number of nodes, $2g - 1$, where g is the number of element groups. In this case, symmetry maximization works exclusively for the objective of finding the most perceptual decomposition. However, without restricting our optimization to the candidate decompositions based on the element groups, the search space would contain trees of different sizes. Symmetry maximization would then work for both of our objectives: most perceptual and simplest explanation. In Section 6, we show findings from such an “optimization alternative.”

4 Symmetry measure

In this section, we first define integral symmetry, a continuous symmetry measure applicable to a *standalone* box abstraction. We then define the symmetry score which forms the objective function of our hierarchical decomposition; see Section 3.4. The symmetry score evaluates the symmetry of two substructures resulting from a structural decomposition; it combines the integral symmetries of the two substructures, with proper normalization.

4.1 Integral symmetry of box abstraction

The basic idea of integral symmetry is to sum up *intra-* and *inter-box* symmetries in a box abstraction. It is not measuring global symmetry but takes into account all partial symmetries. Since any two vertically or horizontally translated (empty) boxes are reflectionally symmetric, there is a strong coupling between the translation and reflection symmetries between the boxes. In our work, we consider *reflection* symmetries only, as they allow for a parameterization over the extent of a box abstraction. Taking into account facade content, we note that in most cases, the atomic elements, e.g., a window or balcony, have reflectional symmetries as well, justifying our exclusive focus on reflection symmetries.

To characterize intra- and inter-box symmetries, we define *symmetry profile*, a continuous 1D function defined over the horizontal or vertical extent of the boxes involved. In the following, we only define the horizontal version; the vertical version is similar.

Intra-box symmetry profile. Let B be an axis-aligned box with horizontal extent $[0, w]$, where w is the width of B . We define

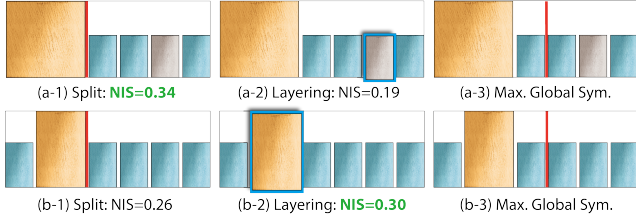


Figure 7: Ranking between split and layering decompositions based on NIS sums. Top: split wins. Bottom: layering wins. (a-3) and (b-3): splits to maximize global reflectional symmetry. Note differences to the best splits as judged by NIS sum.

the symmetry profile, $p_B(x)$, of B , $x \in [0, w]$, as the area overlap between B and the reflection of B about the vertical line at position x . It is easy to see that $p_B(x) = 2xh$ if $x \leq w/2$ and $p_B(x) = 2(w-x)h$ if $x > w/2$, where h is the height of B . The plot of $p_B(x)$ is a hat function; see Figures 6(a-1) and (b-1).

Inter-box symmetry profile. Consider two repeated boxes B_1 and B_2 , where we recall that only repeated boxes can be symmetric to each other. We define the inter-box symmetry profile $p_{B_1, B_2}(x)$ between B_1 and B_2 as the area overlap between B_1 and the reflection of B_2 about the vertical line at position x ; see Figures 6(a-2) and (b-2). $p_{B_1, B_2}(x)$ is parameterized over the horizontal extent of the two boxes. Clearly, the order between B_1 and B_2 here does not matter. We define $p_{B_1, B_2}(x) = 0$ if B_1 is not a repetition of B_2 .

Integral symmetry (Int-Sym). Given a box abstraction (a substructure) S , its integrated symmetry profile is simply the sum:

$$p_S(x) = \sum_{B_1, B_2 \in S} p_{B_1, B_2}(x) + \sum_{B \in S} p_B(x),$$

where x is over the horizontal extent of S . Figures 6(a-3), (b-3), and (d) show a few examples.

To measure the integral symmetry of S , we compute a *weighted* integral where the weight is a Gaussian function centered at the mid-point of S . The weight serves to incorporate *centricity* into the symmetry measure. Specifically, assuming that the horizontal extent of S has been translated into the interval $[-w_S/2, +w_S/2]$ where w_S is the width of S , then the integral symmetry (IS) is

$$\mathcal{I}(S) = \int_{-w_S/2}^{+w_S/2} p_S(x)g(x)dx, \text{ where } g(x) = e^{-\frac{x^2}{(w_S/3)^2}}. \quad (2)$$

With a fixed bounding box, IS takes on the maximum value when S is a single (blank) box. Note that (2) is only the horizontal version of IS definition. We compute a weighted sum of the horizontal and vertical versions of IS to obtain the IS of a box abstraction.

The use of a weight function is to incorporate spatial information into the perception of symmetry. With a constant weight, the IS is unable to differentiate between an AHA pattern and an AAH pattern; it is merely an aggregation of the self-symmetries of the A's and H's and the symmetry between the two A's. Other factors being equal, the visual attention tends to be focused on the center of a visual stimuli [Findlay 1995]. We employ the centricity Gaussian as the weight function to assign more weights to reflection axes closer to the center. This way, AHA would be perceived as more symmetric than AAH; see Figure 6. However, the centricity Gaussian is only one possible way to model visual attention, other factors such as density, color, etc., may also be considered.

4.2 Symmetry score of decomposition

Suppose that a structure decomposition \mathbf{n} produces two substructures S_1 and S_2 . A first attempt at defining the symmetry score at \mathbf{n} is to take the sum $\mathcal{I}(S_1) + \mathcal{I}(S_2)$. However, this measure has a bias towards a trivial decomposition, e.g., with S_1 being a single box for layering or a narrow strip for split. This is easy to see on a blank box with dimension $w \times h$. For simplicity, we ignore centricity and vertical symmetry, then the sum of IS for a split $w \rightarrow w_1 + w_2$ is $(w_1^2 + w_2^2)h/2$, which is maximized when $w_1 = w$.

Normalization. The bias is due to a lack of proper normalization in the IS. Let us denote the tight bounding box of a substructure S by $\beta(S)$. For a split operation resulting in substructures S_1 and S_2 , we define the following normalized version of the IS sum:

$$\mathcal{N}_s(S_1, S_2) = \frac{1}{2} \left[\frac{\mathcal{I}(S_1)}{\mathcal{I}(\beta(S_1))} + \frac{\mathcal{I}(S_2)}{\mathcal{I}(\beta(S_2))} \right]. \quad (3)$$

We call \mathcal{N}_s a *normalized IS sum* or NIS sum. Since for any substructure S , $\mathcal{I}(S) \leq \mathcal{I}(\beta(S))$, \mathcal{N}_s lies in the interval $[0, 1]$.

For a layering operation resulting in top layer S_2 and bottom layer S_1 , we define a slightly different NIS sum:

$$\mathcal{N}_l(S_1, S_2) = \frac{\mathcal{I}(S_1^*) + \mathcal{I}(S_2)}{\mathcal{I}(\beta(S_1)) + \mathcal{I}(\beta(S_2))}, \quad (4)$$

where S_1^* is the box abstraction obtained from S_1 after structure completion (see Section 3.4). The difference to the normalization in (3) is due to the imbalanced roles played by the two layers: $\beta(S_1)$ always contains $\beta(S_2)$. The normalization in (4) also puts \mathcal{N}_l in the interval $[0, 1]$. Both NIS sums alleviate the bias issue with the IS sum mentioned above. For example, the NIS sum for any split or layering applied to a blank box is the constant 1.

Figure 7 compares \mathcal{N}_s , \mathcal{N}_l , and a split strategy based on maximizing normalized *global* symmetry (NGS). We replace IS in (3) by a GS measure where the GS of a box abstraction S is the area overlap between S and its reflection about the center line, horizontal or vertical. The final GS measure takes the maximum of the two.

Symmetry score. We define the symmetry score at node \mathbf{n} in a decomposition hierarchy T by scaling its NIS sum by $\mathcal{I}(\beta(S))$, where S is the substructure stored at node \mathbf{n} . Such a scaling assigns more weights to nodes higher up in T during symmetry maximization. Let S_1 and S_2 be the children of S in T , then

$$\text{SymScore}(\mathbf{n}) = \text{SymScore}(S) = \mathcal{N}(S_1, S_2) \cdot \mathcal{I}(\beta(S)), \quad (5)$$

where $\mathcal{N} = \mathcal{N}_s$ or \mathcal{N}_l , depending on the decomposition type.

5 Applications

A structural understanding enables applications which process or manipulate facade images at a higher and more semantic level than appearance-based approaches. This capability is further strengthened by a hierarchical model. In this section, we develop three applications which strongly utilize our analysis results.

5.1 Structural facade editing

Our structural editing strategy for 2D facades is straightforward. The user is provided with a few options to alter the structural representation we obtain, resulting in structural and geometric changes to the box abstraction. The facade content is retargeted based on the dimensional changes of the boxes. Finally, a facade image is generated via instantiation. The user is free to choose any hierarchy level of the structural representation for structural editing.

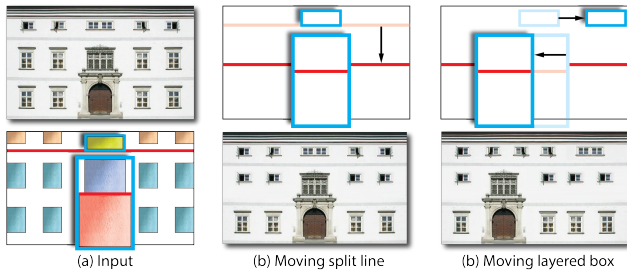


Figure 8: Two structural editing operations. Editing of structural representation and instantiated output facades are shown.

Editing operations. The structural editing application we develop is not meant to be even close to a full-fledged facade editor; that is beyond the scope of this paper. Our goal is to demonstrate the utility of our analysis results. With this spirit, we only implement two editing operations (see Figure 8):

- *Moving a box split line:* This leads to resizing of the two sub-structures about the split line, thus two retargeting operations. We describe the retargeting application in Section 5.2.
- *Moving a layered box:* A top layer may be translated within the confines of the bounding box of the bottom layer. Such a translation would leave holes on the bottom layer that need to be filled. We achieve this by structure completion which relies on the depth order implied by the result of our analysis. Any facade element occluded by a top-layer box is simply deleted, where occlusion relations are determined by the depth order.

Instantiation. To better illustrate the editing results visually, we instantiate all the boxes as well as the wall with image content. First, we instantiate the final wall box by cutting off all the foreground boxes and filling the holes. In our implementation, we execute user-assisted hole filling by copy-and-pasting image patches. Note that this semi-automatic step is performed only once. Instantiation of the foreground boxes is automatic. To achieve that, recall that each foreground box B in the input box abstraction encloses a rectangular patch of the input facade image. The image content in B may be slightly scaled or duplicated (during retargeting). After editing, the image content of all the foreground boxes are simply painted onto the wall in back-to-front order. To obtain such an order, we again rely on the structural representation.

5.2 Facade retargeting

We only consider axis-aligned resizing of boxes and focus on effects of structure retargeting without attempting to resolve all finer-level artifacts that are common to image retargeting. Starting from the root and recursively, as a box is resized, its two child boxes are resized to retain their relative positioning and proportions. Thus facade retargeting reduces to the resizing of *leaf* boxes that contain element groups. Since each element group corresponds to a rectangular grid, retargeting only involves readjusting spacing, box counts, and dimensions to best fit a new grid. To this end, we follow a scheme that is similar to that of [Wu et al. 2010]. Occlusions between layers are handled in the same way as in editing.

Retargeting of irregular facades would not have been possible without a hierarchical organization of the facade elements. Compared to the structure-aware retargeting of Lin et al. [2011], our scheme operates on rectangular grids rather than 1D structures with heavy use of box-to-box alignment. General alignment detection and enforcement during retargeting are both beyond the scope of this paper.

Currently, we require the user to specify pairs of elements that need to be aligned. This is propagated to their element groups. Group-to-group alignment and then box-to-box alignment then follow the scheme described in Lin et al. [2011].

Finally, we allow a group of elements to be *locked* so that they are not scaled. For example, when vertically elongating a facade the user may not want to stretch the first floor with a door. The user scribbles over elements to be locked (see Figure 11). The locking constraints are propagated to the element groups. If both children of a node are locked, the node is locked and the locking constraint is passed up the tree. For a node with only one child locked, retargeting is applied to the unlocked child only.

5.3 Structural facade exploration and retrieval

The facade exploration tool we develop is based on structural representations obtained from our analysis. We pre-analyze a given database of facades to obtain their structural representations. The user sketches a query facade structure, a box abstraction, which is analyzed using our algorithm on-the-fly, producing a query structural representation. The user is allowed to create a global bounding box and other boxes therein (layering), to split a box, to move a split line, and to move a top-layer box. The query is compared to all the stored representations using a tree-to-tree similarity distance.

The tree-to-tree distance is adopted from [Torsello et al. 2005], which is computed recursively based on a node-to-node distance for only the leaf nodes. In our case, the leaves correspond to element groups. To compare two element groups, we first normalize their bounding boxes and then measure the Hausdorff distance between the two sets of boxes.

6 Results and evaluations

Our experiments are conducted on a database of 600 facade images collected from various sources (mostly online photos). Each image is properly cropped and processed into a box abstraction using our interactive tool. It took about 8 hours to process the whole database. The facades vary greatly in structural complexity and irregularity. The whole collection can be viewed in the supplementary material and all data will be made publicly available. This section only presents a sampler of results. For many more results and demonstrations, e.g., on facade editing and interactive exploration, please refer to the video and supplementary material.

Analysis results. A few representative results can be found in Figure 9, where we show the real-world facade, box abstraction, and optimal hierarchical decomposition found. Our analysis is seen to obtain succinct explanations of the underlying facade structures in most cases, especially with the possibility of layering. Results in (a), (b), (d), and (e) are all on globally asymmetric facades with varying degrees of irregularity. Figure 9 (d) shows that symmetry plays the dominant role by grouping seemingly distant elements. Some editing and retargeting results based on these analyses are shown in Figures 10 and 11, respectively.

Parameters. All the analysis results were obtained using the same parameter setting. Weights for horizontal and vertical components of the IS measure are both 0.5. Key parameters for the genetic algorithm can be found in Section 3.4, while the others all take the default setting from the available C++ library GALib.

Statistics and timing. Our analysis algorithm operates on the box abstractions and the optimization is restricted to the candidate decompositions. A typical building facade contains between 5 and 20 element groups. The root node of a tree contains the largest

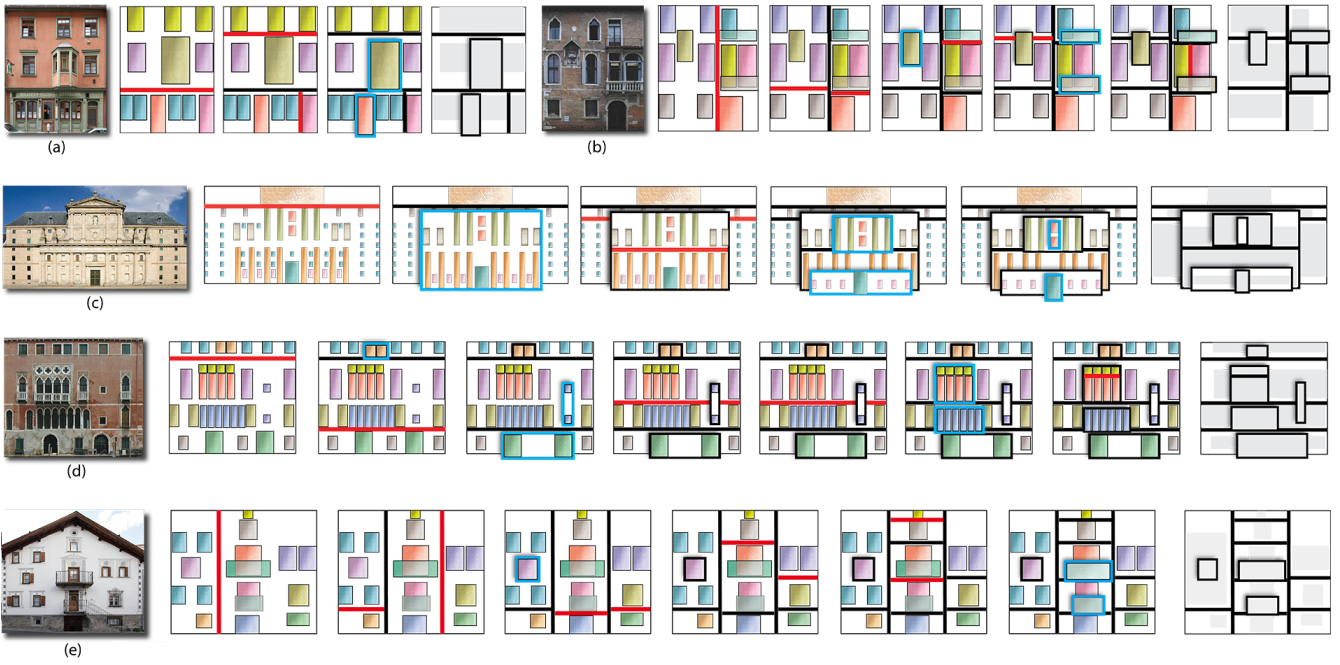


Figure 9: Analysis results on real-world irregular facades (left). The right most image in each set is a collapsed view of the resulting structural representation. The middle sequence shows the optimal hierarchical decomposition obtained. A current split line is shown in red and layered box in light blue border; both colors turn to black in the next level of the hierarchy.

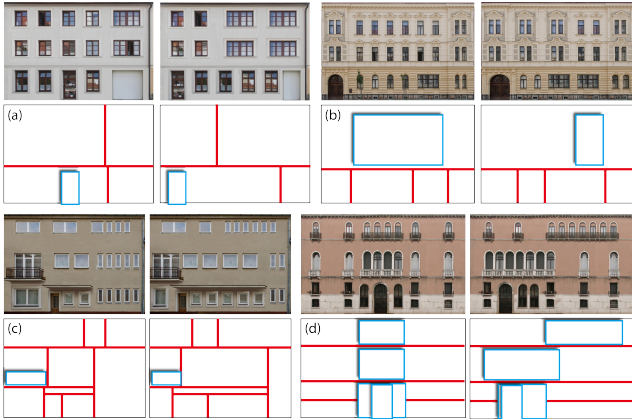


Figure 10: Some facade editing results. Alteration of structural representations and instantiated facade images are both shown.

number of candidate decompositions, which is between 8 and 40, for our dataset. The time consuming aspect of the analysis is the evolution of the tree population. The total analysis time ranges between 20 to 160 seconds over all tests. Structural editing and facade exploration are both performed in real time.

Evaluation of integral symmetry. The integral symmetry (IS) measure plays a key role in our analysis. It is difficult to objectively evaluate the measure — the ultimate goal is to mimic human perception. To this end, we conducted a user study. We prepared a set of *symmetry ranking* tests. In each test, a user is presented with a pair of box abstractions. The user is asked to tell which one of the two patterns is more symmetric and provide a confidence value, on a scale of 1 to 5 (high confidence), for the judgement. All the user study materials can be found in the supplementary material.

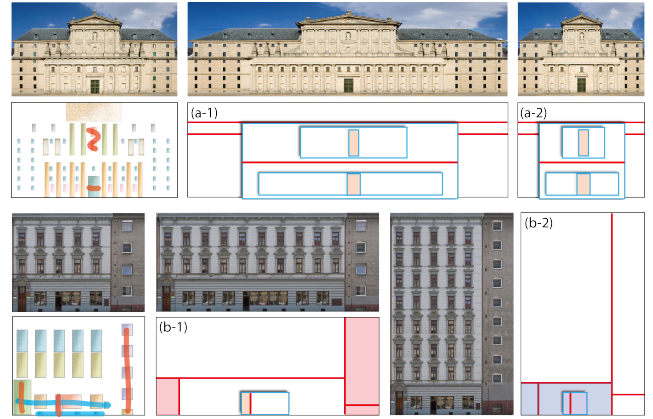


Figure 11: Retargeting results with locking constraints (blue scribble: vertical locking; red scribble: horizontal locking).

We measure the confidence-weighted accuracy score as the percentage of *winning* pattern pairs on which the IS ranking agrees with the user symmetry ranking. One winning pair contributes $c/5$ to the computation of the score, where c is the user-reported confidence value for that pair. We collected ranking results on 46 pairs of box patterns from 15 users. The accuracy score obtained is **88.4%**, which positively demonstrates the potential of the measure.

Evaluation of symmetry-driven decomposition. Objectively evaluating our symmetry-driven analysis is also difficult. Hence we rely on user feedback again. We had considered asking participants to examine full trees which represent the decompositions, but the task proved to be too demanding. The current study breaks the problem down and asks the user to examine individual decompositions and compare our algorithm to two other alternatives. The first



Figure 12: Facade retrieval results on two queries using three different methods. Top three returns are displayed. Note that our tree-to-tree distance is invariant to left-right switching.

is to find the split which maximizes the normalized global (reflectional) symmetry measure (NGS), as described in Section 4.2. The second scheme is an adaptation of graph-cut segmentation (GCS) to box abstractions. It is important to note that in this comparison, we allow *arbitrary* splits beyond the candidates selected based on element groups, for all three methods. Detailed explanation of NGS and GCS schemes can be found in the supplementary material.

In the study, each user is presented with a series of binary-choice questions. Each question asks the user to choose one of two decompositions of a query facade substructure (randomly chosen from those tested in our experiments) that represents the “*best high-level explanation*” of the substructure. One of the two choices is the decomposition based on our symmetry score and the other is the result from either GCS or NGS. The study starts with a detailed introduction to structural decomposition and the two operations, split and layering, with examples. We measure the success of our approach by counting the percentage of user returns where the user chose our decomposition — this counts as a “win” for our symmetry score.

The study collected data from 45 participants. They returned answers to a total of 600 questions each comparing our method to GCS and NGS, respectively. Against GCS, we obtain a winning percentage of **73.2%** and against NGS, we obtain **79.4%**. In our studies, 75% of the participants are computer science teachers and graduate students, between 20 and 50 years of age. Some had experiences with image processing research. The others are frequent computer users with varying careers and backgrounds.

Facade retrieval. We use our structural representations as queries for structure-driven facade retrieval. Figure 12 shows a sampler of results, compared with results using two state-of-the-art appearance-driven retrieval algorithms that had been applied to facade images: one based on SIFT [Lazebnik et al. 2006] and one based on micro-structure descriptor (MSD) [Liu et al. 2011]. Evidently, our representation is able to retrieve structurally similar results which may differ in finer-level image attributes.

Unnatural result. Some unnatural results, such as the split in Figure 13(b), may be produced by our analysis. Placing more emphasis on horizontal symmetry, 0.8 vs. the default 0.5, leads to what may be perceived as a more natural split in (c). Also, altering the input

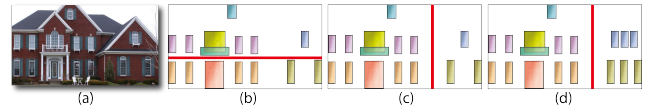


Figure 13: A unnatural split (b) from our analysis on an irregular facade (a). A more natural split (c) can be obtained by placing more emphasis on horizontal symmetry or altering the input (d).

slightly by expanding the grid on the right results in the more natural split. Note that our current IS measure accounts for area overlaps computed for *foreground* boxes only. Hence, a box abstraction having more white space tends to have a lower IS value. A careful perceptual study is needed to understand the interplay between foreground and background on symmetry perception.

Optimization alternatives. As one of the final tests, we check whether restricting our search to the candidate decompositions prevents us from finding a better hierarchy. On 200 randomly chosen inputs, we run the same optimization but allow arbitrary splits — this violates principle P-1 (Section 3.3) and increases the search space. Then we compare the value of the objective function (1) for two solutions. The results show that in 87% of the cases, identical results are obtained, which strongly indicates that the perceptual law of grouping is reinforced by our symmetry measure. The remaining 13% of cases all show minor differences only in the lower tree levels. Note also that genetic algorithms are stochastic and do not always find global optima. Running the algorithm for more iterations improves on the 87%. Finally, we examine improvements made by the genetic algorithm over greedy optimization. On the above 200 inputs, the results show that genetic algorithm obtains higher value of the objective function (1) in 96% of the cases.

7 Conclusion, limitations, and future work

We develop an algorithm for hierarchical and layered analysis of irregular facades, seeking a high-level understanding of the facade structure. The computational approach is a symmetry-driven search for the optimal binary decomposition hierarchy via a genetic algorithm. The analysis result defines a structural representation, which can be utilized for facade structure editing and exploration.

Our work is still only a preliminary attempt at the general analysis problem. Currently, low-level analyses including box abstraction and element grouping are carried out with user assistance. Our analysis is limited to axis-aligned structures and recursive binary decompositions. In cases where a multi-way partitioning is the best explanation, any ordering of a series of binary partitions is likely to be an artificial one. To consider all possible multi-way decompositions would increase the search space significantly, thus additional criteria must be considered to narrow down the search.

Semantics beyond symmetry. We are only utilizing symmetry cues to drive the structural analysis, while being fully aware that symmetry alone does not reveal all the semantic information to disambiguate the search. For example, our symmetry measure characterizes an aggregation while there ought to be cases where prominent local features draw the most attention.

Learning symmetry measures. While the results from our evaluations are quite encouraging, the symmetry measures presented are admittedly still quite limited. For one, the IS measure is only a single numerical value and one piece of information that can be extracted from a symmetry profile. Moreover, the normalization schemes are somewhat ad-hoc without rigorous justification. In reality, there may be a multitude of *perceptual* factors that affect the

perceived symmetry of a structure or goodness of a decomposition. Except for our consideration of centrality, the current definition of IS is quite limited in terms of perceptual considerations; after all, the ultimate symmetry measure is one that best conforms to human perception. We believe that an interesting future work would be to collect the various factors as features and rely on supervised learning to discover their effects on symmetry perception.

Future work. The most desirable pursuit would be to incorporate repetition detection and element grouping into the current optimization framework. Our analysis is not restricted to 2D facades. We would like to apply it to document layouts or 3D architectures. Finally, our approach is purely symmetry-driven and designed to analyze facade structures, not decorative or ornamental architectural styles. Investigations into how to model generic architectural styles and incorporate domain knowledge from architects into the analysis are both interesting avenues for future work.

Acknowledgments. We would like to thank all the reviewers for their valuable comments and feedback. Thanks should also go to Yangyan Li and Niloy Mitra for their helpful discussions. This work is supported in part by grants from NSERC Canada (611370), NSFC China (61202333, 61025012 and 61232011), Guangdong Sci. and Tech. Program (2011B050200007), Shenzhen Sci. and Inno. Program (CXB201104220029A and KC2012JSJS0019A), CPSF (2012M520392), and the Israel Science Foundation.

References

- ALIAGA, D. G., ROSEN, P. A., AND BEKINS, D. R. 2007. Style grammars for interactive visualization of architecture. *IEEE Trans. Vis. & Comp. Graphics* 13, 4, 786–797.
- BOKELOH, M., WAND, M., AND SEIDEL, H.-P. 2010. A connection between partial symmetry and inverse procedural modeling. *ACM Trans. on Graph* 29, 4, 104:1–104:10.
- BOKELOH, M., WAND, M., SEIDEL, H.-P., AND KOLTUN, V. 2012. An algebraic model for parameterized shape editing. *ACM Trans. on Graph* 31, 4, 78:1–78:10.
- BUFFART, H., LEEUWENBERG, E., AND RESTLE, F. 1983. Analysis of ambiguity in visual pattern completion. *Journal of Experimental Psychology: Human Perception and Performance*, 9980–1000.
- CHAO, Y., TIAN, H., LONG, Q., AND TAI, C.-L. 2012. Parsing facade with rank-one approximation. In *Proc. of CVPR*.
- FINDLAY, J. M. 1995. Visual search: eye movements and peripheral vision. *Optometry and Vision Science* 72, 461–466.
- GRAHAM, J. H., RAZ, S., HEL-OR, H., AND NEVO, E. 2010. Fluctuating asymmetry: Methods, theory, and applications. *Symmetry* 2, 2, 466–540.
- HARADA, M., WITKIN, A., AND BARAFF, D. 1995. Interactive physically-based manipulation of discrete/continuous models. In *Proc. of SIGGRAPH*, 199–208.
- HOCHSTEIN, S., AND AHISSAR, M. 2002. View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, 5, 791–804.
- KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2004. Symmetry descriptors and 3D shape matching. *Symp. on Geom. Proc.*, 115–123.
- LAZEBNIK, S., SCHMID, C., AND PONCE, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. of CVPR*, 2169–2178.
- LIN, J., COHEN-OR, D., ZHANG, H., CHENG, L., SHARF, A., DEUSSON, O., AND CHEN, B. 2011. Structure-preserving re-targeting of irregular 3D architecture. *ACM Trans. on Graph* 30, 6, Article 183.
- LIU, G.-H., LI, Z.-Y., ZHANG, L., AND XU, Y. 2011. Image retrieval based on micro-structure descriptor. *Pattern Recognition* 44, 9, 2123–2133.
- MARTINET, A. 2007. *Structuring 3D Geometry based on Symmetry and Instancing Information*. PhD thesis, INP Grenoble.
- MITRA, N. J., PAULY, M., WAND, M., AND CEYLAN, D. 2012. Symmetry in 3D geometry: Extraction and applications. In *Proc. of Eurographics STAR Report*.
- MUSIALSKI, P., WONKA, P., ALIAGA, D. G., WIMMER, M., VAN GOOL, L., AND PURGATHOFER, W. 2012. A survey of urban reconstruction. In *Eurographics State-of-the-art Report*.
- PALMER, S. E. 1977. Hierarchical structure in perceptual representation. *Cognitive Psychology* 9, 4, 441–474.
- PARISH, Y. I. H., AND MÜLLER, P. 2001. Procedural modeling of cities. In *Proc. of SIGGRAPH*, 301–308.
- PAULY, M., MITRA, N. J., WALLNER, J., POTTMANN, H., AND GUIBAS, L. 2008. Discovering structural regularity in 3D geometry. *ACM Trans. on Graph* 27, 3, 43:1–11.
- PODOLAK, J., SHILANE, P., GOLOVINSKIY, A., RUSINKIEWICZ, S., AND FUNKHOUSER, T. 2006. A planar-reflective symmetry transform for 3D shapes. *ACM Trans. on Graph* 25, 3, 549–559.
- SHEN, C.-H., HUANG, S.-S., FU, H., AND HU, S.-M. 2011. Adaptive partitioning of urban facades. *ACM Trans. on Graph* 30, 6, 184:1–184:9.
- SIMARI, P., KALOGERAKIS, E., AND SINGH, K. 2006. Folding meshes: hierarchical mesh segmentation based on planar symmetry. *Symp. on Geom. Proc.*, 111–119.
- STAVA, O., BENES, B., MECH, R., ALIGA, D., AND KRISTOF, P. 2010. Inverse procedural modeling by automatic generation of L-systems. *Computer Graphics Forum (Eurographics)* 29, 2, 665–674.
- TEBOUL, O., KOKKINOS, I., SIMON, L., KOUTSOURAKIS, P., AND PARAGIOS, N. 2011. Shape grammar parsing via reinforcement learning. In *Proc. of CVPR*.
- TORSELLO, A., HIDOVIC-ROWE, D., AND PELILLO, M. 2005. Polynomial-time metrics for attributed trees. *IEEE Trans. Pat. Ana. & Mach. Int.* 27, 7, 1087–1099.
- WANG, Y., XU, K., LI, J., ZHANG, H., SHAMIR, A., LIU, L., CHENG, Z., AND XIONG, Y. 2011. Symmetry hierarchy of man-made objects. *Computer Graphics Forum (Eurographics)* 30, 2, 287–296.
- WERTHEIMER, M. 1923. Untersuchungen zur lehre von der gestalt. ii. *Psychologische Forschung* 4, 301–350.
- WONKA, P., WIMMER, M., SILLION, F., AND RIBARSKY, W. 2003. Instant architecture. *ACM Trans. on Graph* 22, 3, 669–677.
- WU, H., WANG, Y., FENG, K.-C., WONG, T.-T., LEE, T.-Y., AND HENG, P.-A. 2010. Resizing by symmetry-summarization. *ACM Trans. on Graph* 29, 6, 159:1–10.