Erschienen in: Computer Graphics Forum ; 37 (2018), 6. - S. 188-204
https://dx.doi.org/10.1111/cgf.13321

Sketching in Gestalt Space: Interactive Shape Abstraction through Perceptual Reasoning

J. Kratt[1], T. Niese[1], R. Hu[2], H. Huang[2], S. Pirk[3], A. Sharf[4], D. Cohen-Or[5] and O. Deussen[1]

[1]University of Konstanz, Germany
{Julian.Kratt, t.niese}@gmail.com, oliver.deussen@uni-konstanz.de
[2]Shenzhen University, China
{ruizhen.hu, hhzhiyan}@gmail.com
[3]Stanford University, USA
soeren.pirk@gmail.com
[4]Ben-Gurion University of the Negev, Israel
asharf@gmail.com
[5]Tel Aviv University, Israel
cohenor@gmail.com

Abstract

We present an interactive method that allows users to easily abstract complex 3D models with only a few strokes. The key idea is to employ well-known Gestalt principles to help generalizing user inputs into a full model abstraction while accounting for form, perceptual patterns and semantics of the model. Using these principles, we alleviate the user's need to explicitly define shape abstractions. We utilize structural characteristics such as repetitions, regularity and similarity to transform user strokes into full 3D abstractions. As the user sketches over shape elements, we identify Gestalt groups and later abstract them to maintain their structural meaning. Unlike previous approaches, we operate directly on the geometric elements, in a sense applying Gestalt principles in 3D. We demonstrate the effectiveness of our approach with a series of experiments, including a variety of complex models and two extensive user studies to evaluate our framework.

Keywords: computational geometry, modelling, interaction techniques, interaction, level-of-detail algorithms

1. Introduction

Reducing and simplifying 3D shapes while keeping their structural essence has been a challenge for artists, architects and cartographers for a long time. This inspired many researchers to develop computational models and representations as powerful means for guiding the observers attention to specific features and for expressing information effectively (e.g. [MZL*09, YK12, Wil11]). Many technical applications such as 3D printing or level-of-detail rendering benefit from geometric simplification because they require data often in a specific resolution or complexity. Abstracting a model while maintaining its semantic structure (structure-aware shape processing) is arguably one of the fundamental problems in shape modelling research [MWZ*13, BCBB15].

In this paper, we introduce an interactive method that allows users to easily sketch abstractions of complex models. Shape abstractions are meant to be observed by humans and judging the aesthetic qualities of an abstracted model are virtues that belong to humans or more precisely, to artists. Thus, we have to involve the human in shape abstraction and cannot leave it to a fully automatic process. An abstraction tool must be intuitive and easy to use, while guiding the user into producing coherent geometric models that maintain the perceived structure of the original. Our key idea for realizing this is to employ well-known Gestalt rules, which allow to maintain the form and overall patterns of such simplified shapes as perceived by humans. The challenge here is to combine the users' intent and Gestalt rules together in a computational framework. In a nutshell, a user expresses his intent by sketching over the 3D model. Our system interprets these sketches using their underlying geometric context, thus narrowing down the space of possible abstractions considerably. Then, it generalizes the detected Gestalt groups by applying a concise series of 3D abstraction operations (Figure 1). Features such as visibility of group elements and whole groups will

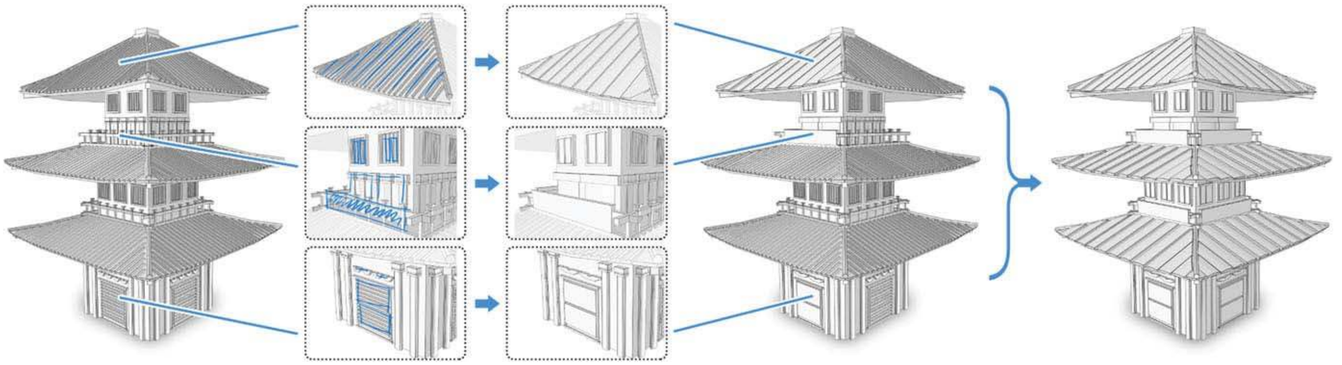


Figure 1: *User-assisted abstraction of a Japanese house. The user sketches his intention on parts of the object. The system automatically finds Gestalt groups based on the loose scribbles and abstracts these groups accordingly. By automatically propagating abstractions to similar geometric parts, the whole model is abstracted (right).*

be used to resolve conflicts between different applicable Gestalt rules and for selecting proper abstraction operations.

Nan *et al.* [NSX*11] used Gestalt rules for the automatic 2D abstraction of façades. In contrast, we apply such rules on 3D elements while accounting for structural characteristics of the input model as well as for its visual perception. As mentioned above, we do not aim for an automatic abstraction, but to assist the user in his interactive abstraction operations. This defines a novel operational domain which we denote as 'Gestalt space'. It is the abstract space that employs user-defined sketches to simplify 3D shapes while maintaining the constraints defined by Gestalt rules. Metaphorically, the user 'sketches in Gestalt space', triggering a series of Gestalt-based operations on 3D objects. Results of our method are abstracted geometries that can be used for a number of applications. Abstraction can be performed in a way that the output can be printed in the given resolution of a 3D printer. The resulting models can also be created in a requested geometric complexity with respect to Gestalt-based perception, while the overall model characteristics are preserved. We demonstrate our technique on models of buildings and technical artefacts. Our main contributions are:

- The formalization of a new operational domain, denoted as 'Gestalt space', which assists users to abstract and simplify complex 3D models while maintaining their structural essence.
- A novel user interface that combines perceptual rules defined by Gestalt principles with 3D sketches that capture the users' intent.
- A framework that computes shape abstractions in real-time and thereby provides immediate feedback to efficiently operate even on complex shapes or on entire scenes.
- Abstraction results of a variety of 3D models and an effectiveness evaluation through two extensive user studies.

2. Related Work

A large number of works perform abstraction based on geometric properties. Attene *et al.* [AFS06] approximate a 3D model with a set of simple primitives. This is done by hierarchical face clustering of the input mesh followed by an automatic fitting of optimal shapes to the clusters. Mehra *et al.* [MZL*09] abstract 3D shapes with a set of characteristic 3D curves and contours. Their method processes the input model in two stages: first, a closed envelope surface is generated as an approximation, then a hierarchical curve network is extracted. The network is used to reconstruct an abstract version of the input model, where fine details on the surface are smoothed out. In contrast to our approach, their method works on a fixed global scale and does not allow for local adjustments. Each part of the input model is abstracted equally and thus, visual important structures of the model might be lost.

Calderon and Boubekeur [CB17] recently propose a method to automatically generate bounding shape approximations of arbitrary complex meshes based on an asymmetric morphological closing. The method produces shape proxies that are close to the input model, even for the coarsest level of approximation. The user can locally influence the proxy scale, resolution and topology in an intuitive way by applying a brush tool. In contrast to our approach, the method does not account for perceptual important structures.

McCrae *et al.* [MSM11] present a learning algorithm to generate abstractions of shapes based on planar sections. This method focuses on a set of slices that describe the input model in an abstract form. Yumer *et al*. [YK12] make the assumption that there is no single abstraction for one object and present a co-abstraction method that generates identity preserving, mutually consistent abstractions for shape collections. The models in the resulting shape collection are abstracted to the maximum extent while maintaining their distinguishing characteristics.

Kada [Kad06] introduces a method for automatic generalization of 3D building models by remodelling the input shape based on half spaces. Forenberg [For07] uses scale-space theory: faces of the input model are moved against each other until 3D features of a certain scale are removed. Grabler *et al.* [GASP08] extend this in order to automatically simplify the visual appearance of building models in tourist maps. The authors rectify the input model using a grid structure, decompose it into different parts and perform a facet shifting strategy to build the simplification. Chen *et al.* [CLLW13] train a neural network with cartographers' expertise and knowledge about constructing ground plane simplifications in order to model a map generalization process. Mitra *et al.* [MWZ*13] provide a comprehensive overview of the field of structure-aware geometry processing.

Symmetries and regular structures. Mitra *et al.* [MGP06] identify partial and approximate symmetries in 3D models by pairing sample points on the surface mesh which have the same local shape signature. Clusters of these matches in a proposed transformation space indicate symmetries in the input model. Pauly *et al.* [PMW*08] present a method for identifying regular and repeated structures in 3D geometry by analysing pairwise similarity transformations. Detailed overviews on symmetry detection are given by Liu *et al.* [LHOKG10] for images and Mitra *et al.* [MPWC12] for 3D geometry.

Gal *et al.* [GSMCO09] introduce iWires, an analyse-and-edit approach to manipulate an existing 3D model while maintaining its characteristics. Wires are extracted from the input model and enhanced with information about geometric features and relations to other wires. The user can edit these wires while the system maintains wire features and relations. In contrast, our method employs freehand sketches that offer more flexibility compared to predefined sets of handles. Recent efforts concentrate on exploiting symmetries for shape manipulation tasks either through hierarchical grouping [WXL*11] or by guiding the deformation and fitting of templates [KWW*14]. Fu *et al.* [FCS*16] perform structure-aware editing of man-made objects through capturing group-specific priors. Their system allows to produce shape variations and structure-aware editing in real-time.

Even more recently, Nishida *et al.* [NGDA*16] leverage the effectiveness of user-defined sketches to guide the automated assembly of snippets of procedural grammars as building blocks to turn sketches into realistic 3D models. Unlike the previous approaches, our method aims at leveraging Gestalt rules as fundamental means for perceptual reasoning for shape abstraction and simplification.

Gestalt-based abstraction. While the above mentioned works perform simplification and abstraction mostly based on geometrical aspects, Nan *et al.* [NSX*11] use high-level Gestalt laws (see also [Wer23] and [Wer38]) to automatically simplify line drawings of architectural buildings. They describe their scenes by a proximity graph connecting elements to their neighbours. Each Gestalt law yields weights for edges forming a multi-label graph cut problem. While their system was an inspiration for us, our problem domain is in 3D and therefore lacks a straightforward definition of Gestalt rules. Emerging from 2D to 3D exposes a vast number of new challenges, e.g. in resolving occlusion, conflicts in group dominance and ambiguities in matching user sketches to groups, which have not been addressed before. Furthermore, we aim at an interactive Gestalt-based abstraction rather than a fully automatic process. With a few simple strokes, the user can freely and easily guide geometric abstraction operations that respect both structural and perceptual groupings within the input 3D shape.

Zhang *et al.* [ZDCW13] and Wang *et al.* [WZMD15] use the Gestalt laws of similarity and proximity to abstract 2D foot-prints of buildings for urban abstraction. Such laws are used for assisting human sketching [LHH*13], for the selection of elements [XFAT12], or for simplifying sketches [LWH15] (law of closure). Some authors highlight the importance of using Gestalt laws also for geometric abstraction [LUB*13, DCNP14, KEP*14]. Most of the existing methods for shape simplification only consider the generalization of line drawings or object compositions in 2D space. In contrast to these techniques, our approach focuses on the abstraction of 3D objects by employing an interactive feedback-loop based on Gestalt principles.

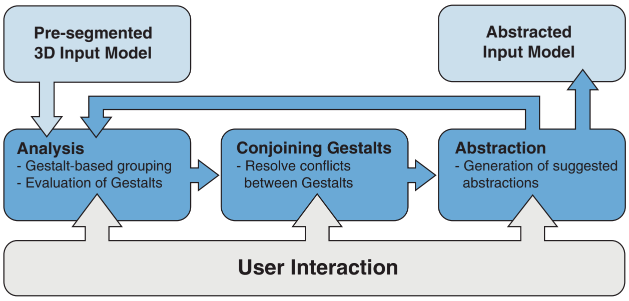


Figure 2: *System overview: a 3D model is analysed and potential Gestalt groups are precomputed. Based on user sketches, groups are selected and the model is abstracted accordingly.*

3. Overview

Our interactive system interprets user sketches and seeks for a sequence of Gestalt-based abstractions that best matches the users' intent. The output is a series of abstractions, each of which is an abstraction of the original model that considers Gestalt principles while preserving characteristic features of the 3D input.

Figure 2 presents a high-level overview of our method. We assume that the input 3D model is already segmented into low-level elements [LSYJ13]. Our method first analyses the input model in terms of Gestalt principles, which are regarded as rules for visually grouping low-level elements into larger aggregated structures. Some of these rules can be quantified, namely similarity, proximity, continuity, closure and regularity. Each rule forms independent (Gestalt) groupings.

The challenge is to resolve possible conflicts among the groupings while identifying the groups that match the users' intent—different Gestalt principles can be applied on the same shape. To resolve such conflicts, we formulate the grouping as an optimization problem (cf. [NSX*11]) and introduce an objective energy function, which encapsulates the characteristics of the Gestalt groups and the users' intent. By minimizing the energy, conflicts are relaxed and unique Gestalt groups are identified.

A fundamental problem of applying Gestalt-rules in 3D is *visibility* (Figure 3). The arrangement of 3D objects might form a Gestalt group but some of the objects are occluded by other surrounding objects. To overcome this limitation, we also consider the visibility of objects when resolving group conflicts in 3D by introducing two novel visibility-related measures, denoted as *group dominance* and *element visibility*.

Both visibility terms are integrated in the objective function for the optimization to favour visually more dominant groups in the

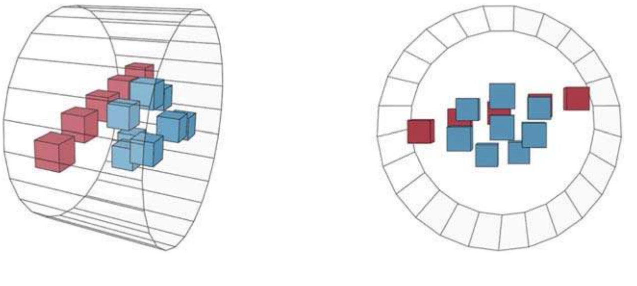


Figure 3: *Visibility affects Gestalt formation: given two Gestalt groups in 3D (left), one group may be occluded by the other under certain viewpoints and thus will not be visible as a group anymore (right). The surrounding cylinder is only rendered to provide a better spatial orientation.*

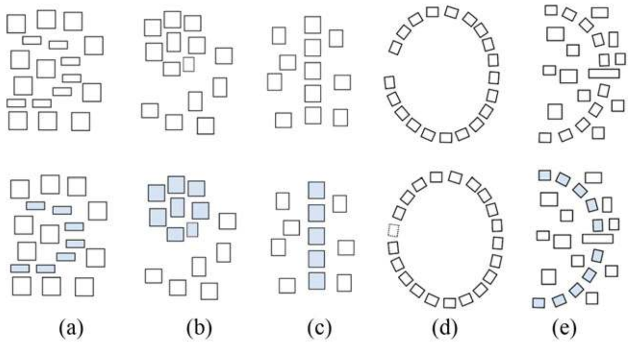


Figure 4: *Gestalt principles: (a) similarity, (b) proximity, (c) regularity, (d) closure and (e) continuity.*

abstraction process. Moreover, we use the visibility terms to determine the degree of abstraction applied to the groups. As we are interested in preserving visually prominent features, visible groups are abstracted more conservatively compared to partially covered or hidden groups.

Group dominance values reflect whether objects are seen as Gestalt groups from multiple views. We perform occlusion analysis and compute the average dominance for each group, which describes its relative importance. Element visibility expresses the extent by which an element is occluded by other elements in its vicinity, computed through ambient occlusion [ZIK98].

Next, we abstract the resulting groups by using one of three possible operations: (i) we create embracing objects in 3D (convex hulls, alpha shapes); (ii) we perform a visual summarization, where a large number of similar objects in a group are represented by a subset of these objects that are potentially scaled; (iii) or we substitute the group by a planar object that shows some engravings of group objects as a form of bas relief (see also Figure 8). The visibility terms will alter what form of abstraction is suggested for a group: occluded groups are simplified more significantly (e.g. by embracing objects) compared to fully visible ones.

For the abstraction, it is not required to apply these operators explicitly. The user directly sketches over the 2D projection of the model to indicate his intent about the abstraction. The idea is that the resulting model simplification follows the user-defined sketches as closely as possible. For this, the system maps the sketches into the Gestalt space by assigning them to the Gestalt groups. The interpretation may not always have a single solution. To resolve these ambiguities, the system presents the user with a gallery of plausible solutions (Figure 9). Once a group is selected by the user, it can automatically be transferred to similar configurations of elements within the model. Here, the user sketch only provides minimal guidance for the model abstraction.

4. Grouping Principles

Gestalt principles describe how humans tend to perceive arrangements of elements and thereby provide a fundamental means for perceptual reasoning about shape abstraction and modification. Unlike previous methods, our approach aims at providing perceptually plausible abstractions of 3D shapes. Sketching in Gestalt space allows to abstract shapes while maintaining their key visual features; we simplify what is perceptually not important. While psychologists differentiate a large number of such principles, our geometric simplification operations focus on similarity, proximity, regularity, continuity and closure (cf. [NSX*11]). Note that such principles only describe groupings seen by the viewer; they do not provide us with concrete operations how to simplify them.

Figure 4 demonstrates how humans perceive arrangements of shapes by using Gestalt principles of similarity, proximity, regularity, closure and continuity. In particular, shapes of the same form are often perceived as distinct groups (Figure 4a), whereas proximity and regularity also form clusters of shapes (Figures 4b–c). The principle of closure describes the tendency of humans to complete a simple shape that is only shown in parts, while the continuity principle states that we continue the directions of shapes in the most simple way. Since both principles *add* content, we do not apply them for geometric abstraction. We do, however, apply a special variant of the law of closure by replacing a group of elements with a base shape that serves as its simplified representation.

In many cases, more than one Gestalt principle applies to the same set of elements, e.g. one set of elements might form a proximity group and another, overlapping set, might form a repetition group. In this case, the geometric configuration determines the predominant principle. To find all potential Gestalt groups, we extend the 2D Gestalt rules to 3D. More specifically, we build a proximity graph G that connects the 3D elements of our scene to their direct neighbours similar to [NSX*11] and then try to find Gestalt groups. For each element p_i, we find its k-closest neighbours, p_j, and connect them with an edge e_{ij}. The edge is associated with a weight $d(p_i, p_j)$ that is related to the Hausdorff-distance $d_H(p_i, p_j)$ between the elements:

$$\begin{aligned} d(p_i, p_j) &= \max\{d_H(p_i, p_j), d_H(p_j, p_i)\}, \\ d_H(p_i, p_j) &= \max_{v_i \in p_i}\{\min_{v_j \in p_j}\{\|v_i - v_j\|_2\}\}, \end{aligned} \qquad (1)$$

where v_i and v_j are vertices of elements p_i and p_j, respectively. Note that the edges of this graph connect the actual 3D elements of our pre-segmented input model. The closest vertices on those two elements (the ones that define the Hausdorff-distance from Equation (1)) are used for building the edges. While *proximity groups* are detected by finding connected elements in G with distances (edge weights) below a given threshold t_p, we detect *similarity groups* by employing a similarity measure that compares the shapes of 3D objects [BKS*05] and try to find groups with a similarity between all their elements that is larger than a given value t_s. We identify *regularity groups* in 3D by finding paths in G that have a regular pattern. A path q that represents a regularity Gestalt in 3D is defined by a sequence of edges $(e_0, e_1, \dots, e_n)$ in G, where the edge lengths vary only to a small extent:

$$\bigcup\{q_i\} \quad | \quad \frac{1}{n}\sum_{j=0}^{n} |\|e_j\| - \bar{e}| < t_l, \tag{2}$$

where q_i are the elements of the regularity group and $\bar{e}$ is the average edge length of the path. Furthermore, the angles α_i and α_{i+1} between two successive edges along the path should only have a small variation as well:

$$\min\left(\frac{\alpha_i}{\alpha_{i+1}}, \frac{\alpha_{i+1}}{\alpha_i}\right) < t_a. \tag{3}$$

More specifically, every node of the proximity graph corresponds to a 3D element of the input model, thus, we assign the centre defined by all vertices of an element to the corresponding node. Therefore, an edge has an orientation in 3D space and we can compute angles α_i and α_{i+1} between successive edges along the path. The thresholds t_l and t_a are used to control variations of lengths and angles that can occur between two successive edges. In addition of being arranged regularly, all elements within the group have to be similar. We find such paths by picking a node of the graph and check for all incident edges if it is possible to start a path in this direction. We keep track of all paths by labelling the corresponding edges.

To account for rotations within a regularity group, we store the main axis of shape elements in the corresponding nodes. To evaluate such axis, we apply principle component analysis (PCA) [Jol86] on the vertices of each element. The PCA determines the directions with the highest variances of vertices, which are used as main axis of an element. Given these axes, we can compute the rotation needed to map one element onto the other. For regular structures, the rotations required to map consecutive elements onto each other are static. While this method seems to be sufficient to detect all regular structures in our scenes, more evolved techniques such as Pauly *et. al*. [PMW*08] may be used to identify more complex patterns.

5. Gestalt and Abstraction in 3D

A fundamental difference between abstractions in 2D and 3D is that the viewpoint in 3D plays an eminent role in perceiving shapes and groups. For example, objects can have large distances in 3D, while appearing as a proximity group from a certain viewpoint (Figure 5). Similar ambiguous and view-dependent effects can be demonstrated for all Gestalt principles. Some of these effects only occur for specific views, which are called as *accidental views*, as opposed to *generic views* that show objects and groups from a standard viewpoint [Bie87].

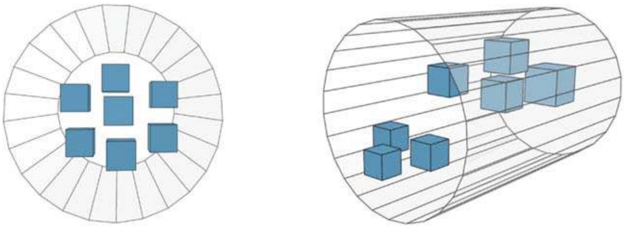


Figure 5: *Distant objects in 3D are sometimes seen as proximity groups through perspective projection. The surrounding cylinders are only rendered to provide a better spatial orientation.*

Since our goal is to create *object* abstractions, we are not interested in accidental views. Instead, we want to find Gestalt groups in generic views. Therefore, we have to define conditions under which a group of elements appears as Gestalt in such views. In the following, we introduce two terms to evaluate the visibility of each Gestalt group: *group dominance* and *element visibility*. The terms are used for abstraction and reflect the visual importance of groups under generic views. We integrate the visibility properties into the optimization to resolve conflicts as well as to determine the amount of abstraction.

5.1. Group dominance

3D Gestalt groups may be visually more dominant than others based on their location with regard to other groups. A group is considered to be dominant if it is perceived as a Gestalt group from multiple views. In order to evaluate the dominance for a view point, we consider the projected area of the entire group and the sum of the projected areas of all individual elements of the group. Our idea is that if all elements of the group are clearly perceivable from the given view point, this also applies to the Gestalt. Further, we consider occlusion caused by other elements which lowers the perception of the Gestalt. We sample the sphere around a group to get a set of view directions $\mathcal{D}$, compute the associated visibility values and obtain an average dominance value per group. This allows us to quantify what we considered as being *visible under generic views*. Figure 6 visualizes the group dominance of two conflicting Gestalt groups.

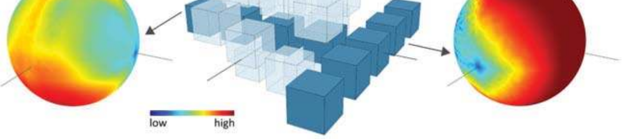


Figure 6: *Visualization of group dominance: we sample the sphere around a group and compute from which directions the group is visible. This defines group dominance. Here, the visibility of two conflicting regularity groups (coloured in dark blue) is blocked by surrounding elements (light blue). The corresponding spheres are shown on each side.*

More specifically, we apply Poisson disc sampling directly on the sphere to sample the set of view directions $\mathcal{D}$. In all our experiments, we chose a minimal geodesic distance of 0.1 between sample points. For a given view $\vec{v} \in \mathcal{D}$, we first estimate how many elements of the group are visible without considering other objects in the scene. We compute the ratio between the projected area of the entire group K and the sum of all projected areas of individual elements $\{\kappa_i\}$:

$$\rho_{\vec{v}} = \frac{area(K)}{\sum_{i=0}^{n} area(\kappa_i)}, \tag{4}$$

where $area()$ is the projected area of the elements and the group without considering any other objects in the scene. A value of $\rho_{\vec{v}} = 1$ indicates no occlusion within the group. If the value is close to zero, elements of the group are highly occluded and therefore the corresponding Gestalt is hardly perceivable. To evaluate the influence of other objects, we determine how much of the group is visible while considering all other elements in the scene. We compute the ratio between the projected area of the group K with and without considering these elements as:

$$\sigma_{\vec{v}} = \frac{area_T(K)}{area(K)}, \tag{5}$$

where T is the set of all objects in the scene without elements of K and $area_T(K)$ is the projected area of group K when considering the occlusion caused by T. Similarly, $\sigma_{\vec{v}} = 1$ if there is no occlusion caused by other elements in the scene; a value close to zero indicates that occlusion is getting serious. The dominance value for a given view direction $\vec{v}$ is given by: $\tau_{\vec{v}} = \rho_{\vec{v}} * \sigma_{\vec{v}}$. Small values of τ indicate that the group is less visible from the given direction (caused by self-occlusion or through occlusion from other objects). Finally, we compute the average dominance value for group K:

$$\bar{\tau}_K = \frac{1}{|\mathcal{D}|} \sum_{\vec{v} \in \mathcal{D}} \tau_{\vec{v}}. \tag{6}$$

5.2. Element visibility

In addition to the group dominance, we also evaluate the visibility for each of its elements. Even if a group has a large group dominance, some of its elements can be significantly occluded by surrounding objects. This disturbs the visual grouping and prohibits using Gestalt principles for certain views. To capture element interferences, we compute a per element visibility [ZIK98] that influences the data costs for elements in the optimization step:

$$A_{\kappa_i} = \frac{1}{2\pi} \int_{\Omega} V(\vec{\omega}) d\vec{\omega}. \tag{7}$$

$V(\vec{\omega})$ is the visibility function that is either 1, if the object is visible along direction $\vec{\omega}$, or 0 otherwise. The result of A_{κ_i} is in the range [0, 1] and describes how much of the sphere Ω centred at the position of κ_i is covered. We only consider occlusion caused by other objects in the scene, not the occlusion within the group itself.

5.3. Energy function

The average dominance value and the element visibility are used to infer the visual importance of 3D Gestalt groups. These measures are important as they specify which parts of a scene can be abstracted while maintaining the main shape characteristics. If a Gestalt group has large dominance and if its elements are not occluded by surrounding elements, abstracting these elements will affect the perception of the overall scene. Thus, for visual abstraction, we aim for simplifying visible groups first, since these groups convey most information of the model. To account for these effects in our interactive system, we introduce element visibility A_p and average visual dominance $\bar{\tau}_l$ into the energy function defined in [NSX*11]:

$$\begin{aligned} E(f) = &\sum_{p \in P} (1 - A(p, f_p)) \cdot D(p, f_p) \sum_{p,q \in N} V_{p,q} \\ &+ \sum_{l \in L} (1 - \bar{\tau}_l) \cdot h_l \cdot \delta_l(f), \end{aligned} \tag{8}$$

where $A(p, f_p)$ is the visibility value and $D(p, f_p)$ is the data cost for an element p if the label f_p is assigned to it. $V_{p,q}$ is the smoothness cost for two neighbouring elements p and q. The term $h_l \cdot \delta_l(f)$ represents the label cost with L being the entire set of labels. Please note that both visibility terms have to be inverted since the optimization seeks for minimizing the energy function. By weighting the individual costs of elements by their visibility, the data cost term reflects how well elements fit to the assigned Gestalt group. Occluded elements are penalized, whereas visible ones are favoured in the optimization. Similarly, the label cost term favours configurations with only a few and cheap labels. By incorporating the dominance value, the label cost of highly occluded groups will be higher, whereas the cost of visible groups will be lower. Thereby, the optimization favours groups that are visually more important. Detailed definitions of the individual terms of Equation (8) can be found in the accompanying Supplementary Material.

Figure 7 shows how visibility modifies the optimization result. Around a 3D grid of cubes, we placed planes rendered transparently for demonstration purposes. All sides of the grid are covered except the front side. The planes enclose the interior cubes and block their visibility from most view directions. Without considering the visibility, the optimization will combine the groups in the interior due to their proximity (b). If we consider the visibility of objects, the visible exterior is instead selected as a group and will be abstracted first (c). This will subsequently force all other (parallel) groups to be

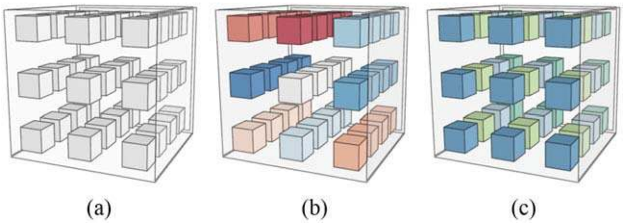


Figure 7: *Effects of dominance and visibility: (a) simple setup of a 3D grid of cubes covered by surrounding planes (transparent). All sides of the grid are covered except the front side. Results of the graph cut-based optimization without considering the visibility (b) and with the modified energy function (c). Elements that belong to the same Gestalt group have the same colour.*

abstracted consistently and thus will propagate through the whole grid.

6. User-Assisted Abstraction

Using the aforementioned Gestalt-based 3D principles, our goal is to employ user sketches and to derive an appropriate abstraction sequence. The Gestalt groupings during this process define a space, and we call it the *Gestalt space*, as the basis for the abstraction process. In the following, we introduce our interactive tools used for abstraction, different operations that we apply to achieve the corresponding abstractions and how user intent is incorporated into the optimization.

6.1. Operations for 3D abstraction

Abstraction is achieved by applying a sequence of Gestalt-based group simplifications. We implemented the following abstraction operations that are either selected automatically by the system based on the geometric configuration of a Gestalt group and its visibility or they are interactively selected based on the user sketch.

Embracing objects. If objects are close together, i.e. their distance is small in comparison to the overall group extent, we abstract them by creating an embracing object (Figure 8a). This operation can be performed by utilizing an (axis-parallel) bounding box, a convex hull or alpha shape [AEF*95]. It is also possible to use more complex simplification methods such as presented by Mehra *et al.* [MZL*09].

Visual summarization. Larger groups of repeating elements (e.g. $n > 20$ by default) are visually summarized by a smaller number of elements that can additionally be scaled to match a given resolution criteria, e.g. for 3D printing (Figure 8b).

Base shape substitution. Sometimes groups of repeating small elements cannot be scaled enough to match a given resolution criterion, because it could lead to excessive distortions of the geometry. In this case, we employ a specialized version of the Gestalt principle of closure. If a group is mostly defined along a plane, we determine the hull of the group and replace it with a plane. The original elements are then merged with this plane, similar to a bas-relief (Figure 8c). Thereby, we keep important details of the original model while creating a larger object with less nuanced details (Figure 14).

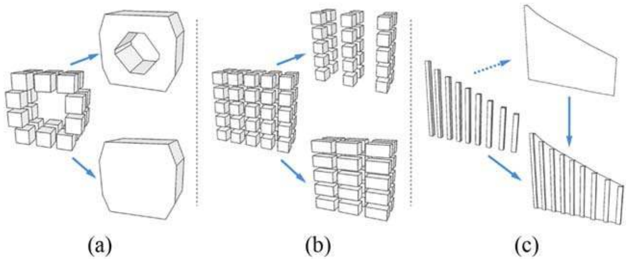


Figure 8: *Abstraction operations: (a) embracing object: a set of cubes can either be abstracted by an alpha shape or its convex hull; (b) visual summarization: a set of cubes can be abstracted by reduced number of elements with additional scaling; (c) base shape substitution: a set of long objects which is substituted by a base plane can be abstracted by engraving parts of the original surface into the base plane, which allows to keep the impression of the original geometry.*

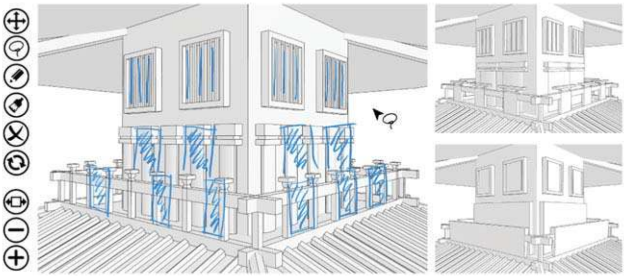


Figure 9: *Our system interface. The user sketches an abstraction over the projected view of the input model. Different interaction modes such as selection or sketching can be used, abstraction results can be refined (scaling/changing the number of visual representatives, etc.), see buttons on the left. Two possible abstractions are shown on the right for the user to select.*

6.2. Interactive tools

Gestalt grouping tools. Initially, the system performs the described Gestalt group optimization (Equation (8)) on the input model. By hovering the mouse over the model, the user can see the groups and is able to split and join them by a cutting and a lasso tool. This enables the user to use his semantic knowledge of the object to direct the abstraction and to resolve ambiguities during the optimization. Furthermore, this helps to achieve user preferred styles, such as a preference for vertical elements.

We implemented a sketch-based interface that shows the input model and provides different interaction possibilities. The user can adjust the view to find the most appropriate view point and can sketch an abstraction. Figure 9 shows a screenshot of the user interface. Based on the sketch and the current viewpoint, the system determines which groups of the 3D model are affected and generates the corresponding abstraction. Often, more than one possible solution exists. In this case, the proximity graph is copied and the simplification is applied to each configuration. All results are presented to the user, who can select the most favourable outcome to progress with the abstraction.

Sketching tools. Since the abstraction of shapes is a highly subjective task, the user should be able to directly influence the abstraction process. We provide this functionality by allowing the user to express his intent on the model with some simple strokes. In most cases, the user sketch consists of many individual strokes. We consider time-stamps and the proximity of such strokes to build stroke sets. Strokes appearing directly after each other are considered to belong together and are summarized to describe a stroke set. These sets are used to infer the type of abstraction applied to an underlying Gestalt group. Our system is able to interpret different types of sketches. By drawing some space-filling strokes, such as

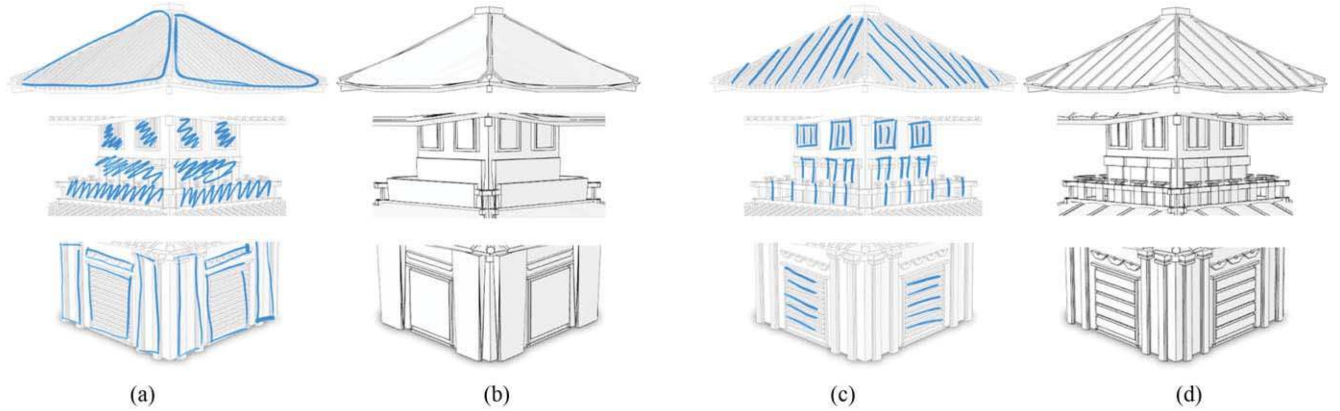


Figure 10: *Abstraction of the Japanese house with distinct sets of sketches. Using closed sketches or zig-zag lines (a) results in abstractions using embracing objects (b). Single strokes (c) instruct the system to use visual summarization (d).*

zig-zag or enclosing lines, embracing objects are used for abstraction (Figures 10a and b). Visual summarization is applied if the user draws individual lines over some regular structure (Figures 10c and d). The number of remaining exemplars of the regular structure then corresponds to the number of stroke sets. To indicate the desired scaling and spacing of the remaining objects, the user can sketch the shape of elements, instead of drawing single lines. By doing this, the system scales the remaining elements accordingly. In case of combining space-filling sketches and individual stroke sets to describe the elements of a regular structure, the base shape substitution operator is applied for abstraction.

Another tool allows the user to transfer the abstraction to other, but similar parts of the model. This is done similar to Xing *et al.* [XCW14] and helps to abstract models more efficiently. We find such structures by employing a graph-isomorphism algorithm [CFSV01], which is extended by also employing 3D orientations. The algorithm finds all isomorphisms in the proximity graph, regardless if their 3D shape is similar or completely different. Therefore, we have to additionally check geometric correspondence within each isomorphism by measuring element-wise geometric distances. After the abstraction of a Gestalt group is done, the user activates the group transfer and the system detects all similar groups, which can then be abstracted similarly and automatically in an efficient way.

Automatic abstraction. Additionally, the user is able to define an entire area to be abstracted automatically. This might involve a number of consecutive abstraction steps and is initiated by a special lasso tool. Please note that the conflict between Gestalt groups is automatically resolved based on our visibility-based optimization; thus, visual important groups are selected for abstraction. Based on the specified 2D area and the geometric configuration of the groups, a sequence of abstraction operations (embracing objects, visual summarization and base shape substitution) are executed and applied to these groups. Here, the abstraction operators are applied based on the type of Gestalt. Proximity and similarity groups will be abstracted by embracing elements, repetition groups will be processed by visual summarization, and if elements of a repetition group form a very thin overall structure, we use a bas relief. Here, we also use the visibility to control the number of remaining elements and their scaling. The higher the occlusion of the regularity group, the more elements are removed. The scaling of remaining elements is adapted in the way that the size of the abstracted group matches the size of the original group.

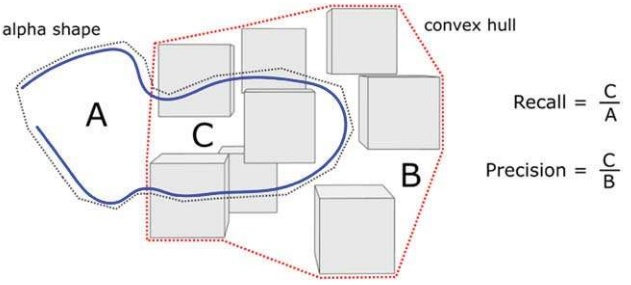


Figure 11: *Precision and recall computation. The convex hull of a proximity group (red, B) is projected onto the canvas. Based on the area of the alpha shape of the sketch (blue, A) and the overlap between the areas (C), we compute precision and recall.*

6.3. Incorporating user intent

Coverage analysis. To identify which groups are intended for abstraction, we consider the coverage of the user sketch and the projected silhouette of the group. For this, we employ the well-known *precision* and *recall* analysis. For each group, we compute a bounding volume depending on the type of Gestalt. For proximity and similarity groups, we use the convex hull of the group elements; for regularity groups, an alpha shape is computed. These bounding volumes are then projected onto the 2D canvas and compared against the alpha shape of the 2D sketch. Figure 11 shows an example for a proximity group and a user sketch that partially covers this group.

Precision and recall provide a measure to identify which group is intended for abstraction. A high precision value indicates that the sketch covers most of the projected area of the group envelope, whereas a high recall value indicates that the entire sketch falls nearly into the projected area. In order to determine which group is intended for abstraction, we compute the F_1 score which combines precision and recall into a single value. This score is defined as $F_1 = 2 \cdot (P \cdot R)/(P + R)$. For abstraction, we then consider the group with the highest score. Please note that the visibility of the selected group has to be taken into consideration, since the group might be occluded by other groups.

Besides detecting the intended group for abstraction, we also use coverage analysis to control the visual summarization operator. We compute the number of stroke sets with a projected area that falls completely into the Gestalt group (recall values close to one). This number is used to infer the number of representative exemplars of the abstracted regularity group. In the next step, we compare the projection of these exemplars with the dimensions of the stroke sets to determine a proper scaling factor for these elements that matches the sketch best, i.e. scales the exemplars to the size of strokes.

Adapting group importance. Only groups with recall values larger than zero are further processed, since for all other groups the projection of the bounding volume is not covered by the user's sketch. However, due to group conflicts, such groups may still 'lose' during the graph-cut optimization and disappear during abstraction. To prevent this from happening, their *importance* value is increased. This is done by temporarily adjusting the data cost term for all elements of the group, in which the user is interested, so that the group will 'win' during optimization:

$$D(p, f_p) := w_c \cdot \min_{f_c \in L_c}(D(p, f_c)). \tag{9}$$

Here, f_p is the label assigned to element p, which is part of the potential group we are interested in. L_c is the set of labels (potential groupings) that are in conflict with f_p. With this adaption, we ensure that the data costs of all elements of group f_p have at least a value of the same costs for all other conflicting potential groups. Since the optimization seeks for minimizing the energy function, groups indicated by the user will 'win' this way. The parameter w_c is in the range [0, 1] and controls the importance. We set $w_c = 0.2$ as default for all our examples.

7. Evaluation

We conducted two user studies to evaluate the automatic 3D Gestalt-based grouping and our proposed sketch-based interface. The participants were both undergraduate and graduate students from different universities. The students are normal computer users without backgrounds in computer science. Additionally, we also provide statistics about the performance of our system.

7.1. 3D Gestalt grouping

In the first user study, we evaluated the efficiency and accuracy of our method to group 3D elements with respect to Gestalt principles. For this, we asked 15 students to manually define Gestalt groups based on how they perceive groups of elements. In total, we showed five input models and their segmentation to the subjects. In Figure 14, the segmentation of two models is shown. By clicking on individual segments, the subjects were able to manually build groups.

Table 1: *Overview of the average time $\bar{t}$ needed by the users to define one group and the average number of groups found per model.*

Model (figure)	#segments	$\bar{t}[s]$	#groups
Japanese house (1)	754	22.98	64.80
Building (16)	1396	7.91	100.67
City (18)	148	17.43	16.13
Bridge (16)	517	11.55	44.87
Eiffel Tower (16)	424	13.63	30.40

We recorded the time and history of applied operations (adding or removing elements to a group) to compare the efficiency. Table 1 summarizes the average timings needed to build one group and the average number of groups that were perceived. Depending on the model complexity, the process of building groups manually takes up to several minutes.

To determine the accuracy, we compute the average F_1 measure over all manually generated groups for a given model with respect to the automatically detected ones. More specifically, for each user-defined group A and automatically generated group B, we can compute the precision (P) and recall (R) value pair in a similar way as in Figure 11, where C is the set of segments that are shared by both groups. To account for the extent of group elements in the computation, we use the volume of segments $vol()$:

$$P = \frac{\sum_{\kappa_i \in C} vol(\kappa_i)}{\sum_{\kappa_i \in B} vol(\kappa_i)} \quad \text{and} \quad R = \frac{\sum_{\kappa_i \in C} vol(\kappa_i)}{\sum_{\kappa_i \in A} vol(\kappa_i)}. \tag{10}$$

Then, the F_1 score is defined as $F_1 = 2 \cdot (P \cdot R)/(P + R)$. Finally, we find the best matching automatically generated group with highest F_1 score for each user-defined group and use the average over F_1 values to compute the accuracy.

Figure 12 shows the distribution of the average values per model. The user-perceived groups match the automatically generated groups well with high accuracy throughout the models. For the building model (Figure 19), we observe a relatively high spread within the average F_1 values. This can be explained by how the students

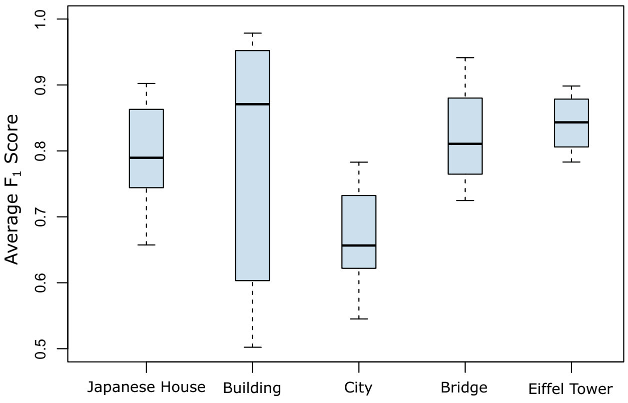


Figure 12: *Boxplots of the average F_1 values per model.*

Table 2: *Performance statistics showing the number of groups #grps, the computation time t_{ig} for the initial grouping, the time t_{ug} needed to update the grouping after changes, the time t_{iso} needed to find similar structures and the computation time t_{op} spent to apply one abstraction operation and the total number #ops of operations. All timings are average values and given in milliseconds.*

Model (figure)	#grps	t_{ig}	t_{ug}	t_{iso}	t_{op}	#ops
Jap. house (1)	286	538	0.93	23.34	80.72	7
Building (16)	1824	2791	1.19	36.24	109.62	4
City (18)	113	52	1.53	2.83	4.65	11
Bridge (16)	206	695	0.31	45.77	215.83	10
Eiffel Tower (16)	320	146	0.03	7.42	41.37	8

build the Gestalt groups. Most students did not build individual groups for each interior window, but instead used one large group. Our system groups elements within each window, which results in low precision values, thus lowering the F_1 score.

In conclusion, our system is able to identify groups of elements automatically that are also perceived by most of the users. Since the grouping of elements is a highly subjective task, we measured subtle differences between manually and automatically generated groups (spread in average F_1 scores). Further, our proposed automatic grouping is performed much faster (Table 2). By exploiting the Gestalt grouping tools (see Section 6.2), the user can adjust the automatically detected groups according to his requirements.

7.2. Sketching interface

We conducted a second user study with 31 students with no experience in modifying or editing 3D shapes to evaluate the effectiveness of our system. For this task, we considered the Japanese house in Figure 10 and the bridge model in Figure 16. For both models, we presented possible abstractions of different parts. Additionally, we showed multiple sets of sketches. The subjects were asked to rate how well a sketch represents a possible abstraction. The score is given in the range of 0–10, where 0 means that a sketch does not fit well to the abstraction and a score of 10 indicates a sketch represents the abstraction very well.

It turns out that the interpretation of user sketches by our system fits very well the expectations of the resulting abstraction. With an average score of 8, most subjects expect a bounding volume for abstraction if the sketch has a closed shape, e.g. Figures 10(a–b). If only single strokes were used, the expectations of the produced abstraction are met with an average score of 6.5, e.g. Figures 10(c–d). Only using zig-zag lines without an enclosing shape was misleading (average score of 4.5). Our system indicates such zig-zag lines in the way that the abstraction is generated using embracing objects.

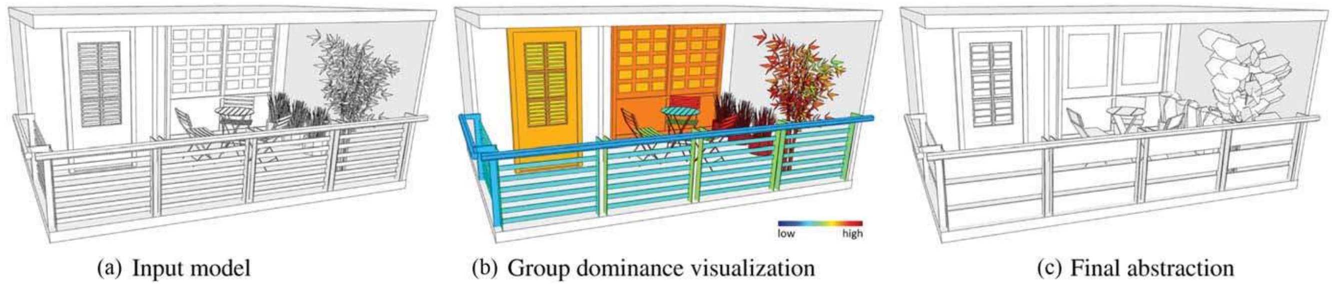
We also invited two people with modelling experience to use our system. The users reported that the interaction concept was considered to be very intuitive and easy to control. For objects with many elements, the selection of groups was identified to be challenging as the system sometimes does not support to focus on selection areas in the projected view. However, once groups were selected properly, the solutions provided by the system in most cases matched the users expectations. The automatic selection of similar groups within complex objects, such as the façade of Figure 16, was considered to be very helpful and efficient.

7.3. Visibility

The mentioned visibility terms (group dominance and element visibility) indicate the importance of elements and groups. We use these terms to resolve conflicts if two or more groups act on the same element and to adapt the form of abstraction used to simplify groups. Please note that for visualization clarity, we omit the user sketches in the following examples.

Figure 13 shows a model of a balcony and demonstrates the usefulness of the integration of visibility computation into the abstraction process. Highly occluded parts (Figure 13b, dark red) are simplified significantly. The strength of abstraction for the plants on the right side of the balcony is reduced successively due to receding occlusion. Also, the abstraction of structures on the windows compared to those on the front door is adapted to visibility. Besides the adaption of the amount of abstraction, we also integrate the visibility terms into the objective function to resolve conflicts between groups in a meaningful way.

Our goal is to abstract parts of the model that are perceived as visually important. This importance is defined by Gestalt groups and our visibility analysis. Since visible groups communicate most of the information of the model, the optimization will favour those groups within the graph cut. Figure 15 shows a temple model and exemplifies how visibility integration affects the abstraction. For



(a) Input model (b) Group dominance visualization (c) Final abstraction

Figure 13: *Abstraction of a 3D balcony model. The strength of the group simplification is based on the visibility, hence the higher the occlusion is the more significant the abstraction is.*

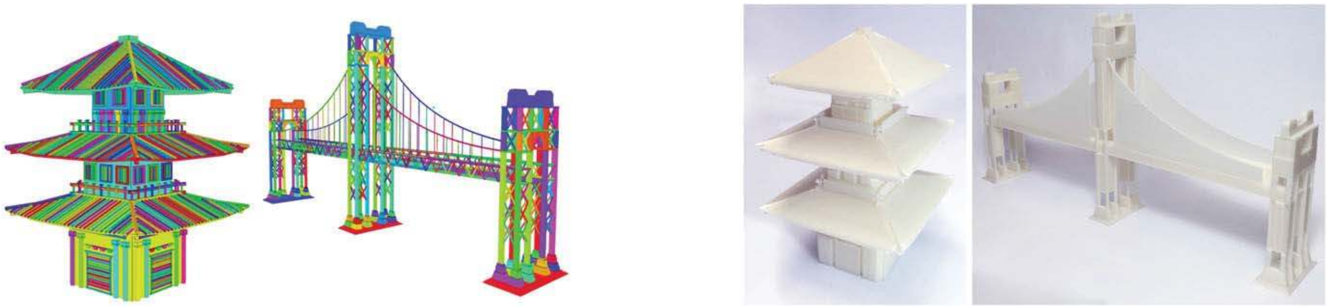


Figure 14: *Segmentation of our input models for further processing and two abstracted models printed in 3D.*

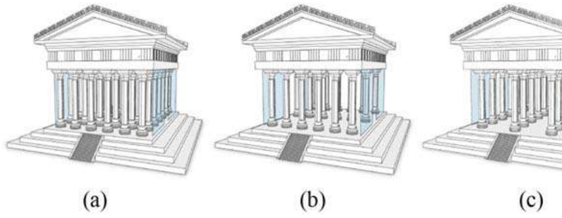


Figure 15: *Effects of visibility. A model (a) is abstracted using the objective function without (b) and with (c) visibility consideration.*

demonstrating purposes, some additional walls (rendered transparently) have been added on the left and right side of the columns. The columns of the temple are arranged on a grid, where each column is part of more than one group. The resulting ambiguities are resolved by the graph cut. Figure 15(b) shows an abstraction result where visibility was not considered. Here, the optimization favours groups that are occluded, which causes an abstraction similar to the original model given in Figure 15(a). In contrast, Figure 15(c) shows the result of an abstraction with our modified objective function. The resulting abstraction reduces the complexity of visible groups. The entire abstraction sequence of the temple model can be seen in the result Section.

7.4. Performance

We measured the performance of our system to demonstrate its usefulness for real-time editing. Table 2 shows computation times in milliseconds for the proposed method. We have implemented our system in C++ on a desktop computer with an Intel i7 processor at 3.2 Ghz and 57 GB RAM. Depending on the model complexity and the number of segments and groups, the overall computation time per interaction is between 41 and 216 ms. The table shows the number of groups the model consists of, the initial time to build the graph and groups, and the average time per interaction that is required to update the groups, to find isomorphisms and the total time per interaction including the simplification.

8. Results

To show the usefulness of our system, we interactively abstract a number of different models. We show manually simplified models and also demonstrate automatic abstraction sequences.

8.1. User-assisted abstraction

In Figure 16, we show a number of user-assisted abstraction operations. In the first row, a user directs the abstraction of a building façade: first, he selects two window frames and scribbles over the vertically repeating elements to substitute them by five bounding boxes (a). By marking the right part of the sketch with two blue outlines, the system is enforced to completely replace the content by bounding boxes. The abstracted version is further simplified (b) by labelling the windows as a whole and subsequently by replacing vertical and horizontal repetitions of windows with bounding boxes (c). In the second row, a complex bridge model with many structural repetitions is processed (e). In Figure 16(f), the user first replaces the truss network with a small number of larger elements (visual summarization) and then replaces the fine strings that attach the road to the bridge with a solid plane using base shape substitution (g).

The resulting model can be printed with a 3D printer (see also Figure 14). Note that 3D printers have physical limitations in their resolution. This requires level-of-detail techniques to explicitly address these constraints and the visual affordance of printed models. We do not claim to provide a more efficient means in terms of the material consumption, but instead focus on maintaining important visual clues that define an object while simplifying it.

The last row in Figure 16 demonstrates that Gestalt-based selection and abstraction goes far beyond conventional processing possibilities. The user draws some steps over a circular staircase to indicate a group and at the same time his abstraction intention. The system finds all similar steps on the stairs and replaces them by a few appropriately scaled steps. In particular, we scale the stairs in such a way that the projected area of the element matches the area of the strokes. Moreover, the bounding volume of the entire model limits the scaling; thus, there is no change in the radius. Similarly, the system abstracts the Eiffel Tower with just few strokes.

Figure 17 demonstrates the effect of dominance and visibility with a small temple (a). As the side view is blocked by the walls, the frontal view is visually more important and thus abstracted first, when the user only scribbles on the frontal part (b). Here, the user also scribbles on the basement of the temple to evoke summarization into two base plates. Note this is not a trivial operation as we have to first compute the Gestalt group and then find a representation of the input base plates by two other plates that are scaled accordingly. The user draws rectangles over the columns to indicate their desired

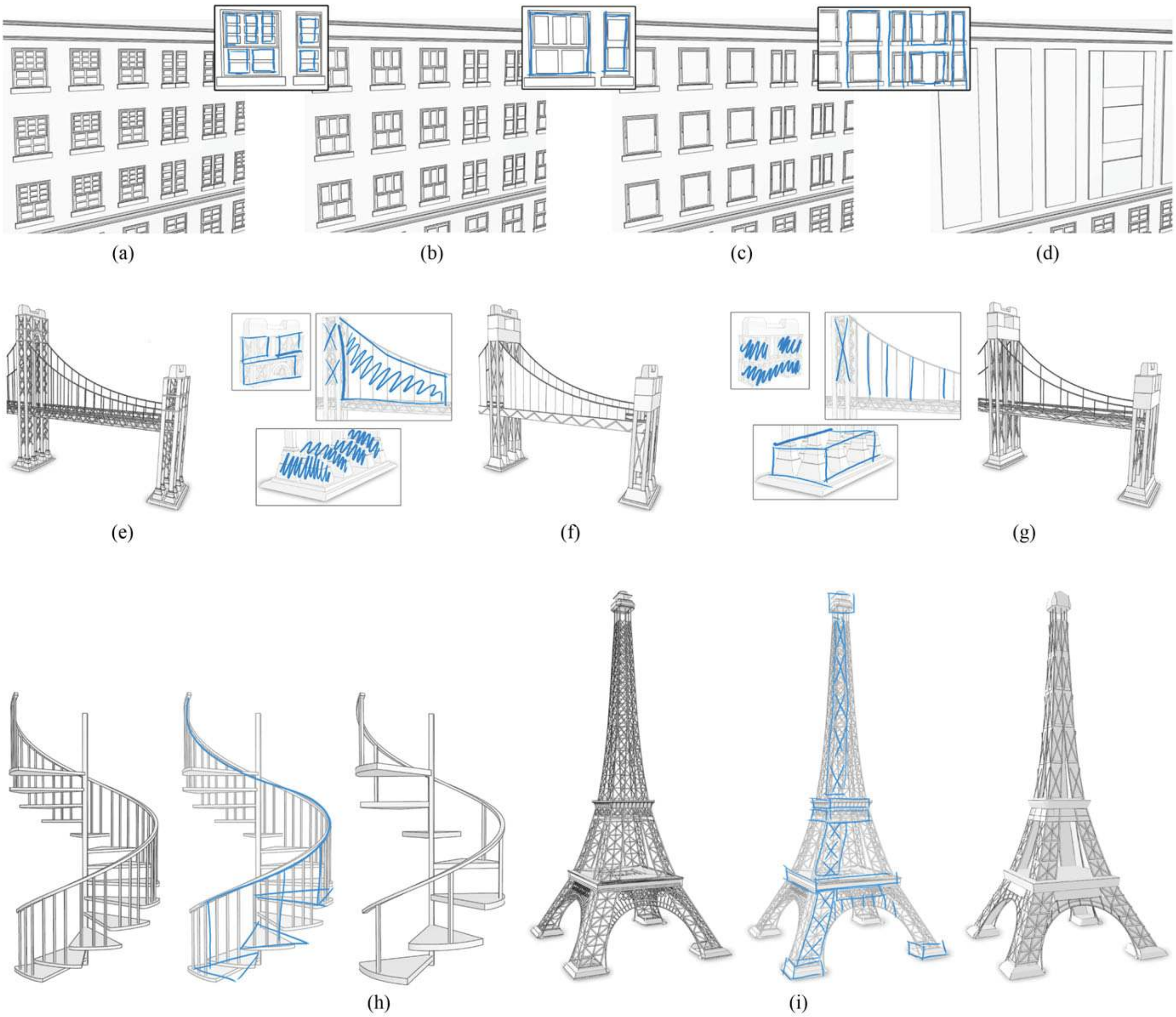


Figure 16: *User-assisted abstraction (descriptions given in the text).*

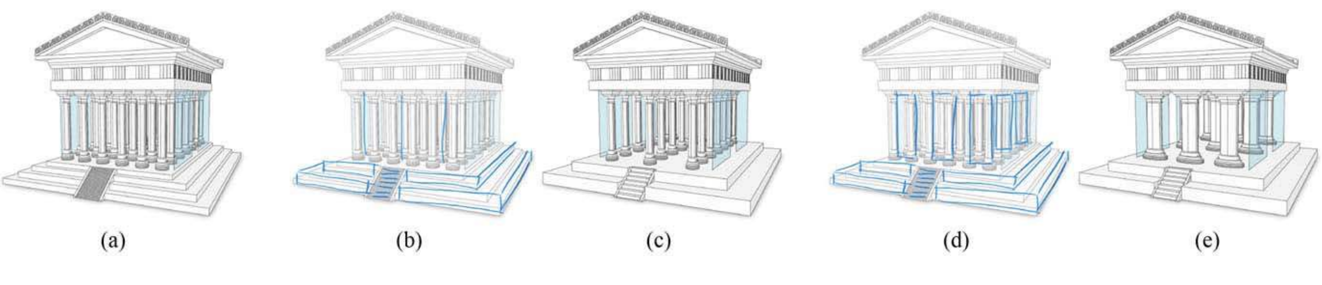


Figure 17: *User-assisted abstraction when visibility is included (descriptions given in the text).*

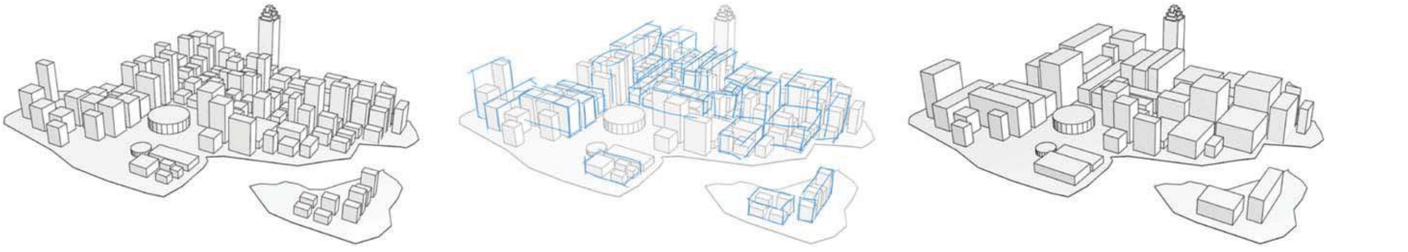


Figure 18: *Abstraction of a city model. With a few strokes, the user is able to combine building models and replace them by embracing objects or visual summarization. This way even complex city models can be processed very efficiently.*

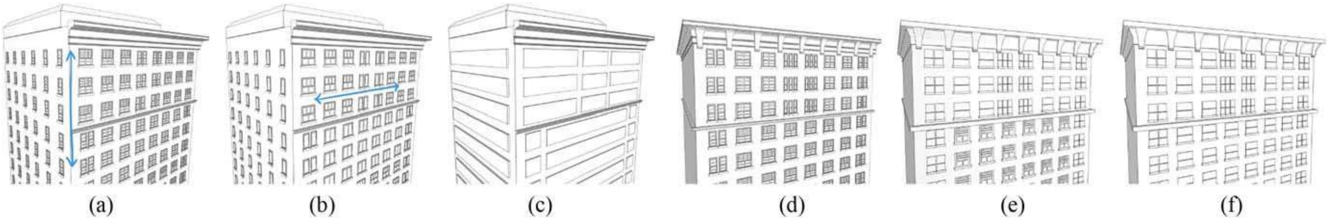


Figure 19: *A building model (a) is automatically abstracted using user-defined directions (indicated by blue arrows). First, the user indicates his preference for vertical abstraction in the first abstraction step (result in b), then in the next step for horizontal abstraction (result in c). Building models (d–f) are the abstraction results produced manually by a professional modelling artist, which are similar to our automatically generated abstractions.*

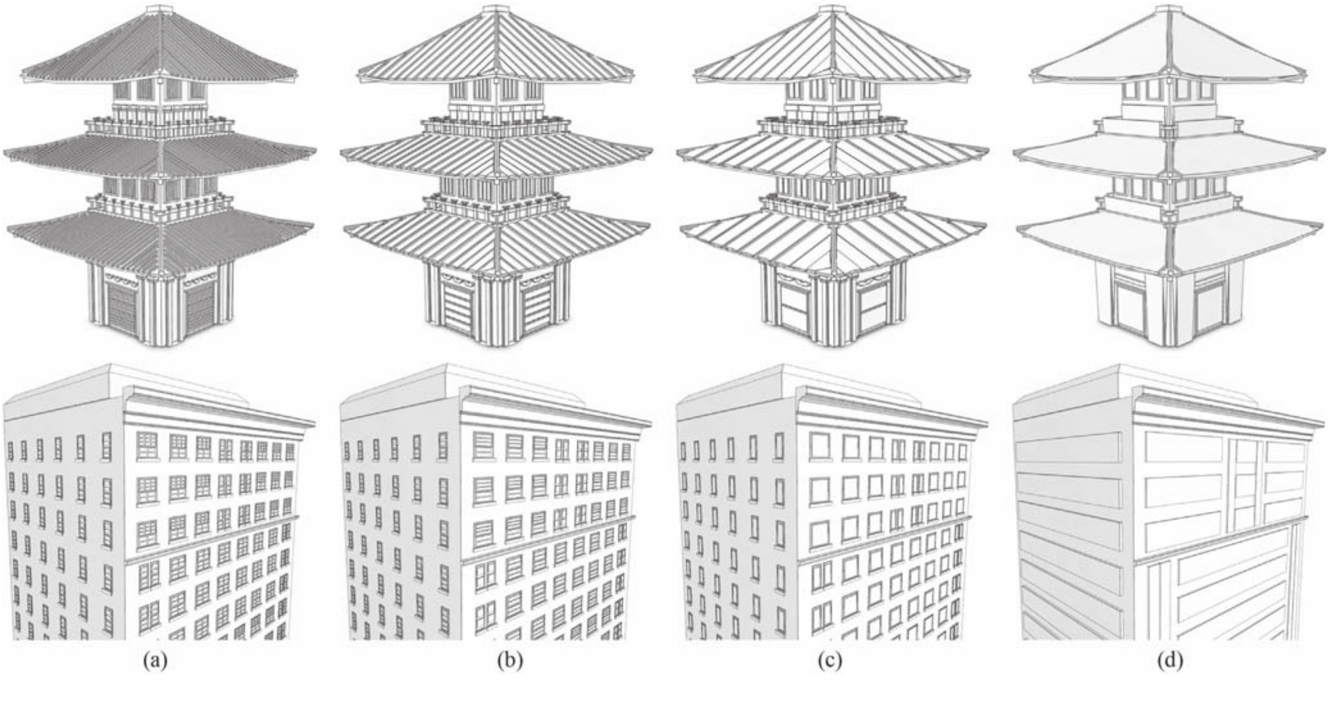


Figure 20: *Automatic level-of-detail sequences. Here, a number of abstraction operations are automatically applied to the input model to match an intended degree of abstraction, i.e. a given number (range) of elements.*

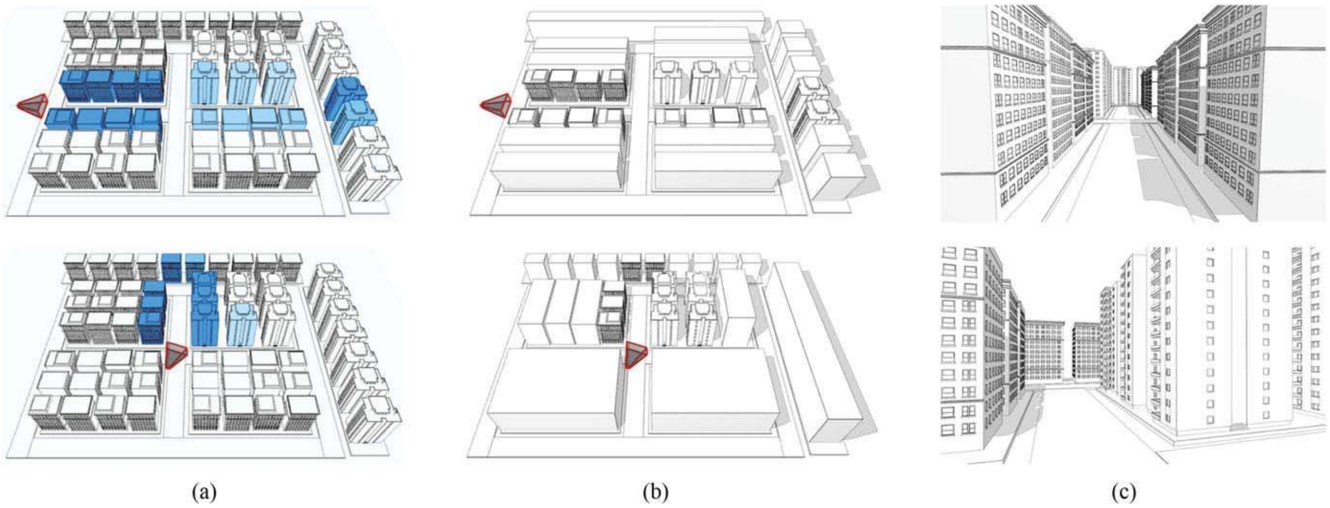


Figure 21: *View-dependent abstraction of a city model. Each row shows the abstraction for a specific viewpoint on the scene. The camera position and orientation of each viewpoint is indicated by the red camera frustum. Based on each view, we compute our visibility terms, which are then used to guide our Gestalt-based optimization and to determine the amount of abstraction. A coloured-coded visualization of element visibility is shown in (b). Buildings that are visible are coloured in blue. The final view-dependent abstractions shown from above and from the perspective of each camera are illustrated in (b) and (c).*

size (d). Furthermore, he also marks columns in rows behind the first visible row. Thereby, he forces the system to abstract all rows in the same way. In Figure 18, we demonstrate that the system can also be applied for the abstraction of city models. Here, the user groups and visually summarizes buildings and whole building blocks.

8.2. Automatic abstraction and level-of-detail

Besides user-guided abstractions, our system also provides an automatic shape abstraction. Here, the user has to define an area of the model which is intended for abstraction. The system then performs a number of consecutive abstraction operations until a user-specific degree of abstraction is established. Thereby, the user forces the system to apply a number of Gestalt abstractions without directing the graph cut and the minimal solution is selected without any user-defined adaptation of the weights.

In Figure 20, we show an automatic abstraction sequence applied on the Japanese house and a building model. For both examples, the user selects the entire model for the abstraction. In every step, we reduce the number of elements by a fixed percentage. Compared to a user-guided abstraction of the same building model (Figure 19), the automatic process ends up with a different abstraction. Nonetheless, it is still a valid result. Figure 22 shows a comparison to the automatic method presented by Mehra *et al.* [MZL*09]. Without considering Gestalt principles, important visual cues would be lost in the abstraction process.

Furthermore, we asked a professional artist to abstract some of the presented models without seeing the results of our system. The task for him was to abstract and group together the elements of the given shape in a semantically meaningful way. Some steps of the automatic abstraction sequence are very similar compared to how an artist would simplify a model. Figures 19(d–f) show the results, which closely match our abstraction results.

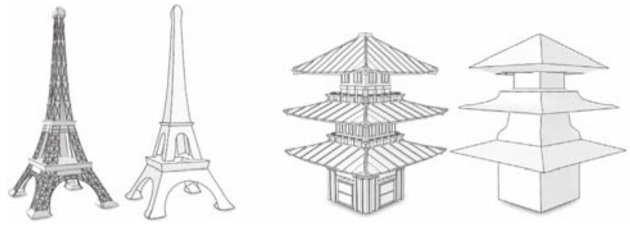


Figure 22: *Models automatically abstracted with the method presented by Mehra* et al. *[MZL*09] (right) in comparison to our abstractions (left). While Mehra* et al. *create very rough approximations, in our case visual important details remain.*

8.3. View-dependent abstraction

Our method can also be applied to abstract models or even entire scenes in a view-dependent way while considering Gestalt principles. To achieve this, we compute our visibility terms (group dominance and element visibility) from a given view on the model. This is different compared to previous results, where we evaluated these terms from the viewpoint of an element or a Gestalt group. This allows us to quantify the importance of elements and groups for a specific view on the model, rather than computing a general importance for *generic views*. Figure 21 shows an example of a city scene, where we applied our view-dependent abstraction. The scene consists of regularly arranged buildings, forming different Gestalt groups. Please note that most of these groups conflict with each other. Each row of Figure 21 shows an automatic abstraction of the scene from a different viewpoint. Based on each view, we evaluate the visibility terms, which are then used within our optimization to resolve conflicts between Gestalt groups and to determine the amount of abstraction. Figures 21(a) and (b) illustrate the colour-coded element visibility and the resulting abstractions shown from above. We use colour coding to indicate highly visible buildings in blue and occluded ones in white. The abstractions shown from the perspective of the cameras are given in (c). By comparing the results of the two viewpoints, it can be seen that conflicts between groups are resolved differently. The optimization favours regularity groups that are mostly aligned with the viewing direction due to higher visibility. This allows us to keep relevant structures in the scene that are perceived from a given view. We also use the visibility to adjust the amount of abstraction applied to groups. While visible buildings remain unchanged, highly occluded ones are abstracted significantly. Even if some parts of the city are completely occluded, we use embracing objects for abstraction, which allows us to keep plausible shadows in the scene. The accompanying video shows a tracking shot for the city model.

8.4. Limitations

Our system has some limitations. For the visual summarization abstraction, we first reduce the number of exemplars of a group and scale the remaining elements. The scaling is performed in the direction of the main axis of the element determined by the PCA. However, if these directions are not aligned with the symmetry axis of the element, the shape might be distorted after scaling. Another limitation regarding the scaling emerges if we want to abstract groups that lie on curved surfaces. Here, the maximum scaling factor of the elements is limited by the curvature. If the scaling is chosen to large, elements might not be attached to the surface anymore.

Although our system is able to simplify nested shape elements that are provided by the segmentation, it is not possible to detect those dependencies automatically with the current implementation. This is shown in Figure 23. We have three plants and the detected Gestalt groups indicated by different colour: one regularity group (red) and three proximity groups (blue). Even though multiple plants would form a regularity group, our system is not able to detect it. Moreover, the connection between two regularity groups that are close to each other cannot be automatically resolved by our system. This might result in a wrong abstraction (Figure 23b). To overcome these problems, we have to rely on the segmentation.

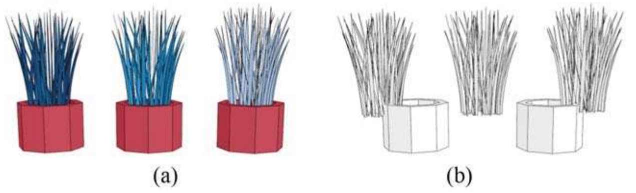


Figure 23: *Failure case: given three plants (a) with the detected regularity group (red) and proximity groups (blue), simplification (b) of the regularity group does not account for scene composition.*

9. Conclusion

In this paper we applied Gestalt principles for the abstraction of complex 3D models. Fully automatic or guided by a number of user sketches, Gestalt principles are applied to elements of the input and visual groups are simplified by a number of operations such as embracing with bounding objects, visual summarization or base shape substitution. We introduced two novel visibility terms to account for the perceptual importance of 3D Gestalt groups. This allows us to resolve conflicts between Gestalt groups in a meaningful way, where visual important groups are favoured for abstraction. Moreover, visibility is used to control the amount of abstraction. We abstracted building models, technical artefacts and a city model. In most cases, our system supports the creation of semantically meaningful abstract representations with only a few user interactions that can be compared to what professional artists will do to abstract shapes. We also showed that our method can be applied to abstract larger scenes in a view-dependent way while still accounting for Gestalt principles.

In this work, we only implemented the most important Gestalt principles, and in the future other, more subtle ones, will follow. We also want to further explore the conceptional space between 2D Gestalt principles and 3D modelling, as we only scratched the surface of possibilities for using Gestalt principles as a means for shape abstraction. Adapting 3D model representations to user perception is a challenging problem for future works in geometric abstraction.

Acknowledgements

This work was financially supported by the grants of DFG (620/20–1), German-Israeli Foundation for Scientific Research and Development (I-1274-407.6), National Science Foundation of China (61761146002, 61522213 and 61602311), Guangdong Science and Technology Program (2015A030312015), Shenzhen Innovation Program (JCYJ20170302153208613 and JCYJ20151015151249564), the Israel Science Foundation (1106/11, 2472/17), the Max Planck Center for Visual Computing and Communication (MPC-VCC) funded by Stanford University and the Federal Ministry of Education and Research of the Federal Republic of Germany (FKZ–01IMC01 and FKZ–01IM10001).

References

[AEF*95] Akkiraju N., Edelsbrunner H., Facello M., Fu P., Mücke E. P., Varela C.: Alpha shapes: Definition and software. In *Proceedings of the 1st International Computational Geometry Software Workshop* (Minneapolis, USA, 1995), pp. 63–66.

[AFS06] Attene M., Falcidieno B., Spagnuolo M.: Hierarchical mesh segmentation based on fitting primitives. *Visual Computer 22*, 3 (March 2006), 181–193.

[BCBB15] Biasotti S., Cerri A., Bronstein A., Bronstein M.: Recent trends, applications, and perspectives in 3D shape similarity assessment. *Computer Graphics Forum 35*, 6 (2016), 87–119.

[Bie87] Biederman I.: Recognition-by-components: A theory of human image understanding. *Psychological Review 94*, 2 (1987), 115.

[BKS*05] Bustos B., Keim D. A., Saupe D., Schreck T., Vranić D. V.: Feature-based similarity search in 3D object databases. *ACM Computing Surveys 37*, 4 (December 2005), 345–387.

[CB17] Calderon S., Boubekeur T.: Bounding proxies for shape approximation. In *Proceedings of SIGGRAPH 2017, ACM Transactions on Graphics 36*, 5 (July 2017), 57:1–57:13.

[CFSV01] Cordella L. P., Foggia P., Sansone C., Vento M.: An improved algorithm for matching large graphs. In *3rd IAPR-TC15 Workshop on Graph-Based Representations in Pattern Recognition* (Beijing, China, 2001), Cuen, pp. 149–159.

[CLLW13] Cheng B., Liu Q., Li X., Wang Y.: Building simplification using backpropagation neural networks: A combination of cartographers' expertise and raster-based local perception. *GIScience & Remote Sensing 50*, 5 (2013), 527–542.

[DCNP14] Dang M., Ceylan D., Neubert B., Pauly M.: Safe: Structure-aware facade editing. *Computer Graphics Forum 33*, 2 (2014), 83–93.

[FCS*16] Fu Q., Chen X., Su X., Li J., Fu H.: Structure-adaptive shape editing for man-made objects. *Computer Graphics Forum 35*, 2 (2016), 27–36.

[For07] Forberg A.: Generalization of 3D building data based on a scale-space approach. *ISPRS Journal of Photogrammetry and Remote Sensing 62*, 2 (2007), 104–111.

[GASP08] Grabler F., Agrawala M., Sumner R. W., Pauly M.: Automatic generation of tourist maps. *ACM Transactions on Graphics 27*, 3 (August 2008), 100:1–100:11.

[GSMCO09] Gal R., Sorkine O., Mitra N. J., Cohen-Or D.: iwires: An analyze-and-edit approach to shape manipulation. *ACM Transactions on Graphics 28*, 3 (July 2009), 33:1–33:10.

[Jol86] Jolliffe I.: *Principal Component Analysis*. Springer Verlag, Berlin, 1986.

[Kad06] Kada M.: 3D building generalization based on halfspace modeling. In *Proceedings of the ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data* (Hanover, Germany, 2006).

[KEP*14] Kratt J., Eisenkeil F., Pirk S., Sharf A., Deussen O.: Non-realistic 3D object stylization. In *Proceedings of the Workshop on Computational Aesthetics* (New York, NY, USA, 2014), CAe '14, ACM, pp. 67–75.

[KWW*14] Kurz C., Wu X., Wand M., Thormählen T., Kohli P., Seidel H.-P.: Symmetry-aware template deformation and fitting. *Computer Graphics Forum 33*, 6 (2014), 205–219.

[LHH*13] Lindlbauer D., Haller M., Hancock M., Scott S. D., Stuerzlinger W.: Perceptual grouping: Selection assistance for digital sketching. In *International Conference on Interactive Tabletops and Surfaces* (St. Andrews, United Kingdom, 2013), ACM, pp. 51–60.

[LHOKG10] Liu Y., Hel-Or H., Kaplan C. S., Gool L. J. V.: Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision 5*, 1–2 (2010), 1–195.

[LSYJ13] Li Q., Sun X., Yang B., Jiang S.: Geometric structure simplification of 3D building models. *{ISPRS} Journal of Photogrammetry and Remote Sensing 84*, (2013), 100–113.

[LUB*13] Løvset T., Ulvang D. M., Bekkvik T. C., Villanger K., Viola I.: Rule-based method for automatic scaffold assembly from 3D building models. *Computers & Graphics 37*, 4 (2013), 256–268.

[LWH15] Liu X., Wong T.-T., Heng P.-A.: Closure-aware sketch simplification. *ACM Transactions on Graphics 34*, 6 (2015), 168:1–168:10.

[MGP06] Mitra N. J., Guibas L. J., Pauly M.: Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics 25*, 3 (July 2006), 560–568.

[MPWC12] Mitra N. J., Pauly M., Wand M., Ceylan D.: Symmetry in 3D geometry: Extraction and applications. In *EUROGRAPHICS State-of-the-Art Report* (2012).

[MSM11] McCrae J., Singh K., Mitra N. J.: Slices: A shape-proxy based on planar sections. *ACM Transactions on Graphics 30*, 6 (December 2011), 168:1–168:12.

[MWZ*13] Mitra N., Wand M., Zhang H. R., Cohen-Or D., Kim V., Huang Q.-X.: Structure-aware shape processing. In *SIGGRAPH Asia 2013 Courses* (New York, NY, USA, 2013), SA '13, ACM, pp. 1:1–1:20.

[MZL*09] Mehra R., Zhou Q., Long J., Sheffer A., Gooch A., Mitra N. J.: Abstraction of man-made shapes. *ACM Transactions on Graphics 28*, 5 (December 2009), 137:1–137:10.

[NGDA*16] Nishida G., Garcia-Dorado I., Aliaga D. G., Benes B., Bousseau A.: Interactive sketching of urban procedural models. *ACM Transactions on Graphics 35*, 4 (July 2016), 130:1–130:11.

[NSX*11] Nan L., Sharf A., Xie K., Wong T.-T., Deussen O., Cohen-Or D., Chen B.: Conjoining Gestalt rules for abstraction of architectural drawings. *ACM Transactions on Graphics 30*, 6 (December 2011), 185:1–185:10.

[PMW*08] Pauly M., Mitra N. J., Wallner J., Pottmann H., Guibas L. J.: Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics 27*, 3 (August 2008), 43:1–43:11.

[Wer23] Wertheimer M.: Untersuchungen zur lehre von der gestalt. II. *Psychological Research 4*, 1 (1923), 301–350.

[Wer38] Wertheimer M.: Laws of organization in perceptual forms. In *A Source Book of Gestalt Psychology*. W. Ellis (Ed.). Routledge and Kegan Paul, Abingdon, UK (1938), pp. 71–88.

[Wil11] Willmott A.: Rapid simplification of multi-attribute meshes. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics* (New York, NY, USA, 2011), HPG '11, ACM, pp. 151–158.

[WXL*11] Wang Y., Xu K., Li J., Zhang H., Shamir A., Liu L., Cheng Z., Xiong Y.: Symmetry hierarchy of man-made objects. *Computer Graphics Forum 30*, 2 (2011), 287–296.

[WZMD15] Wang Y., Zhang L., Mathiopoulos P. T., Deng H.: A Gestalt rules and graph-cut-based simplification framework for urban building models. *International Journal of Applied Earth Observation and Geoinformation 35*, 2015), 247–258.

[XCW14] Xing J., Chen H.-T., Wei L.-Y.: Autocomplete painting repetitions. *ACM Transactions on Graphics 33*, 6 (November 2014), 172:1–172:11.

[XFAT12] Xu P., Fu H., Au O. K.-C., Tai C.-L.: Lazy selection: A scribble-based tool for smart shape elements selection. *ACM Transactions on Graphics 31*, 6 (November 2012), 142:1–142:9.

[YK12] Yumer M. E., Kara L. B.: Co-abstraction of shape collections. *ACM Transactions on Graphics 31*, 6 (November 2012), 166:1–166:11.

[ZDCW13] Zhang L., Deng H., Chen D., Wang Z.: A spatial cognition-based urban building clustering approach and its applications. *International Journal of Geographical Information Science 27*, 4 (2013), 721–740.

[ZIK98] Zhukov S., Iones A., Kronin G.: An ambient light illumination model. In *Rendering Techniques '98* (Vienna, 1998), G. Drettakis and N. Max (Eds.), Eurographics, Springer, Vienna, pp. 45–55.

Supporting Information

Additional Supporting Information may be found in the online version of this article at the publisher's web site: