Co-Abstraction of Shape Collections

Mehmet Ersin Yumer Levent Burak Kara

Carnegie Mellon University

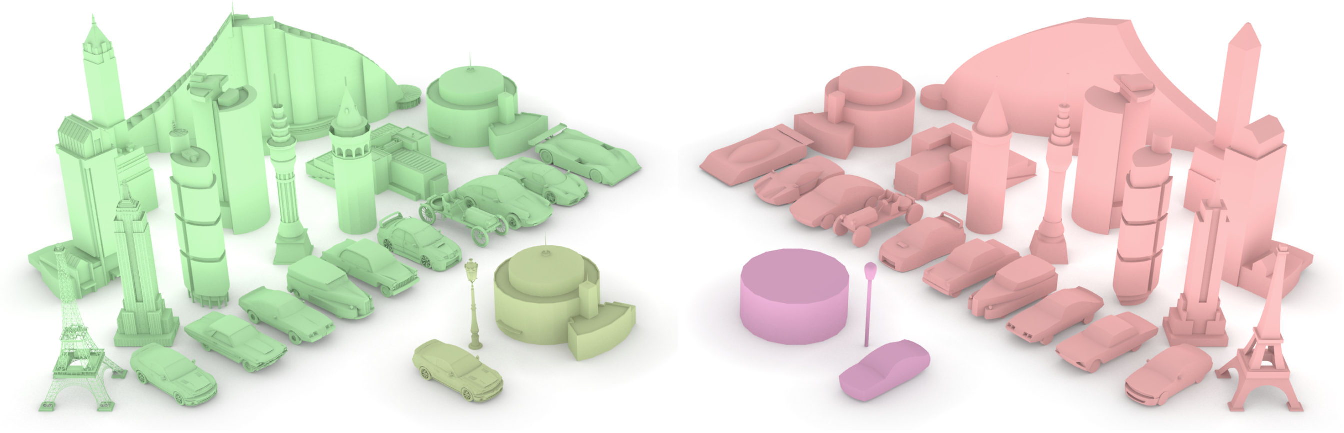

Figure 1: *Given a shape collection, our system creates a mutually abstracted set. We compute a spectrum of abstractions for each model. The most appropriate abstraction level is then determined in relation to other abstractions in the collection. When a collection includes geometrically similar models such as the cars (**green set**), the resulting abstractions (**red set**) preserve details unique to each model. When the original models are dissimilar (**yellow set**), the abstraction of the same models can become much simpler (**magenta set**). (The original models used throughout this paper are the copyright of the respective owners from Google 3D Warehouse, Archive-3D and Artist-3D.)*

Abstract

We present a co-abstraction method that takes as input a collection of 3D objects, and produces a mutually consistent and individually identity-preserving abstraction of each object. In general, an abstraction is a simpler version of a shape that preserves its main characteristics. We hypothesize, however, that there is no single abstraction of an object. Instead, there is a variety of possible abstractions, and an admissible one can only be chosen conjointly with other objects' abstractions. To this end, we introduce a new approach that hierarchically generates a spectrum of abstractions for each model in a shape collection. Given the spectra, we compute the appropriate abstraction level for each model such that shape simplification and inter-set consistency are collectively maximized, while individual shape identities are preserved.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Applications;

Keywords: co-abstraction, mutual abstraction, shape simplification, shape character extraction.

Links: DL PDF

1 Introduction

> "Our experience is not of the real world itself but of an internal representation, a miniature virtual-reality replica of the world." –*Lehar, 2003.*

An object, independent of its true physical form, may manifest itself differently to different observers [Lehar 2003]. As the subject of many philosophical debates for centuries [Shorey 1901], indirect realism has influenced researchers in cognitive sciences and human perception, who have long studied the relation between perception and experience [Pylyshyn 2007; Foster 2000]. Inspired by the strong coupling between the two, we introduce co-abstraction. Instead of searching for a single abstraction of a 3D object in isolation, we hypothesize that an abstraction is meaningful only in the context of other objects. In particular, every object can have a multitude of possible abstract forms, but the most suitable abstraction can be identified by conjointly studying the abstractions of all the objects that make up a collection of models.

This view is based on the previous studies which argue that human cognition system tends to encode only a few characteristic features of a given object, rather than storing every little detail associated with it [Noë et al. 2000]. The amount of information preserved in this way is determined by the observer's knowledge of other objects. Only minimal amount of information is preserved sufficient enough to help the observer discriminate the objects in his or her knowledge base. To this end, we define an abstraction as a geometrically simplified object which preserves the features unique to its shape. From this perspective, our vision is similar to Mehra *et al.* [2009], where an abstraction is sought in the form of characteristic curves generated from the 3D geometry. However, in the quest for the "right abstraction," we believe it is further necessary to generate a spectrum of abstractions, while maintaining the ability to quantify and measure the difference in the perceived level of detail along the spectrum.

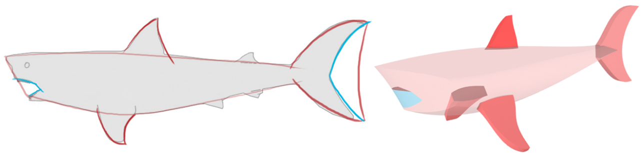

Figure 2: *Artist's 'inside-outside forms' abstraction prior to drawing in 2D (left). Our method follows a similar approach in 3D (right). The light red body represents the current abstraction. Subvolumes identified in multiple layers are shown (darker subvolumes most recent).*

Inspired by the *inside and outside forms* used by drawing artists, we also generate the most abstract form first, and then progressively add further details (Figure 2). In our approach, a detail is represented as a chunk of volume, which can be added or subtracted to the current abstraction to generate a new one. This is analogically similar to the drawing scenario in which a base form is further embellished with a multitude of details introduced successively.

Given a collection of models, our method is based on generating a spectrum of abstractions, from which the most suitable ones are determined using co-abstraction. Our approach is tailored such that all models are abstracted to the highest extent possible while each model preserves its unique geometric characteristics. Moreover, our approach aims to maintain a general sense of consistency in the abstractions such that the results share similar levels of detail. To this end, we first describe an abstraction method that uses volumetric primitives to decompose the original model into successively smaller components. In each step, the identified primitives are geometrically beautified (Section 3.3.1) to closely reproduce the form present in the shape of the original model. This formulation results in a compact representation of the original geometry as a set of implicit surfaces and blending functions. Given a collection, once the spectrum of abstractions are computed for all the models, our co-abstraction algorithm identifies the most suitable abstraction level for each, by minimizing a joint entropy computed from inter- and intra-model geometric relationships.

Contributions. Our primary contributions are as follows:

- A hierarchical 3D shape abstraction method based on subvolumes which generates a spectrum of abstractions for a given input model.
- A co-abstraction scheme where the models in a collection are maximally abstracted, while each model maintains its identity preserving characteristics. This scheme also works to maintain a consistent level of detail within the collection of models.

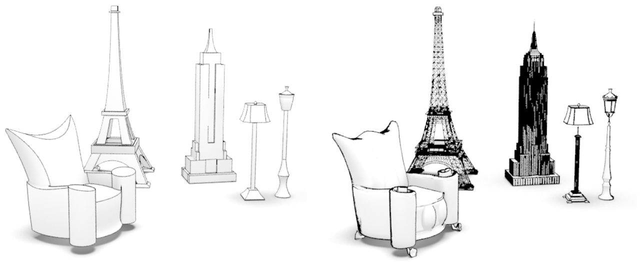

Figure 3: *Pen-and-Ink rendering [Winkenbach and Salesin 1994] of a scene created with our co-abstraction method (left), and with the original models (right).*

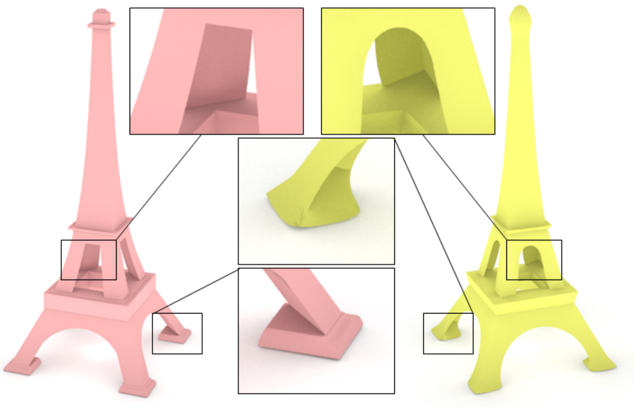

Figure 4: *Left: Our abstraction. Right: Abstraction method of [Mehra et al. 2009].*

2 Related Work

Imagery Abstraction. Imagery abstraction involves the simplification of the form and content communicated in the image to enhance its expressiveness. Driven by this goal, imagery abstraction has been extensively studied in non-photo realistic rendering [DeCarlo and Santella 2002], and shape recognition [Demirci et al. 2009]. Similar to our 3D basic primitive identification, Sala *et al.* [2010] utilizes a set of 2D primitives to determine closed contours for shape recognition. These contours are later combined into simplified regions. Using an approach similar to skeletonization in [Macrini et al. 2008], Mi *et al.* [2009] first organize the shape into parts that attach to the main body, and later simplify the shape via pruning and beautification operations. Their 2D part decomposition and pruning moves the abstraction from detailed to simple, while our hierarchical 3D primitive identification and beautification operates in the reverse direction. In architectural drawing abstraction, Nan *et al.* [2011] propose a consideration of the elements in a scene conjointly, rather than abstracting each object in isolation. Our method shares similar goals in that we seek to abstract out a set of models concurrently such that the overall perceived abstraction level of the collection remains consistent, while individual model identities are preserved. Our approach, however, enables abstractions to be applied directly to 3D data, making downstream conventional rendering methods straightforward to apply (Figure 3).

3D Shape Abstraction and Simplification. Our co-abstraction method involves the abstraction of individual objects, and as such

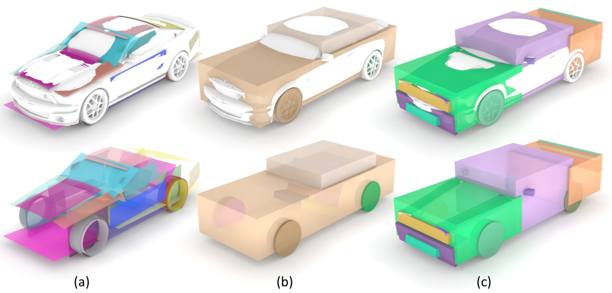

Figure 5: *(a)RANSAC primitives based on surfaces, (b) Volumetric segmentation of [Attene et al. 2006] -the disconnected parts in the input model are manually stitched into a single mesh prior to processing in (b)- (c) Our basic primitives based on volumes.*

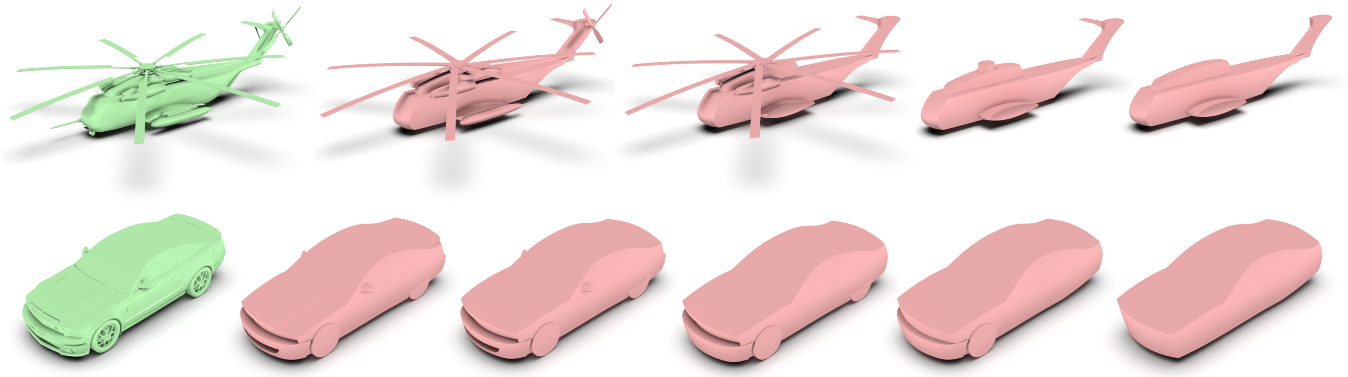

Figure 6: *Abstraction spectra of two models. Only a limited number of abstraction levels are shown for each model. Refer to the supplementary documentation for more detailed abstraction spectra of various models presented in this paper.*

shares similar aspirations with a large body of previous works. Mehra *et al.* [2009] describes an abstraction method that represents a 3D shape with a compact set of feature curves. De Goes *et al.* [2011] uses exoskeletons to simplify the geometry into a curve network similar to Mehra *et al.* [2009], but require a connected mesh topology for its identification. In both cases, interpolating surfaces can be generated from these curves resulting in a simpler representation of the original model. In our approach, the main building blocks of the abstraction are the *subvolumes* (Section 3.2) as opposed to a curve network. This enables us to explicitly control and quantify the differences between various abstractions in a given abstraction spectrum (Section 3.1). Moreover, high quality surfaces that represent the abstraction geometry is a by product of our formulation. As shown in Figure 4, an advantage of this is that it does not require further processing of the extracted curve network, and therefore resulting surfaces and volumes are not influenced by such elements. In [Mehra et al. 2009], for instance, when surfaces are constructed from such edges, undesirable geometric artifacts may arise (curled base feet and mid-section platform on the right side of Figure 4). McCrae *et al.* [2011] generate abstractions in the form of planar cuts through 3D geometries, using a learning algorithm. However, their method focuses on slices as the representation of the abstract form, while our method produces closed volumes.

Our approach to abstraction begins with basic primitive search, which identifies volumetric regions through a randomized search process. It is similar to RANSAC methods [Schnabel et al. 2007; Schnabel et al. 2009; Li et al. 2011] in that our method also does not require an underlying topology. However, RANSAC methods focus on direct surface identification from point sets, and use the encoded surface primitives primarily for this purpose (Figure 5a). Our method on the other hand, takes a volume-centric approach to primitive fitting (Figure 5c). Additionally, once identified, the surfaces of our primitives are further beautified to closely match the original model. One advantage of our approach is that it desirably identifies a representative outer shell of an object, regardless of whether the vertices of the original model are concentrated to the surfaces (*e.g.* coming from scan data), or dispersed throughout its volume (*e.g.* coming from CAD/assembly data with many internal components). Figure 7 demonstrates this idea.

3D Shape Segmentation. Our hierarchical abstraction of a single model can be viewed as a volume-based segmentation method. Several previous methods also explored volume-based segmentation [Lien et al. 2006; Kreavoy et al. 2007; Attene et al. 2008]. Most closely related to our work, we compare our method to that described in [Attene et al. 2006], as shown in Figure 5. There are three aspects that render our approach more suitable for geometric abstraction; (1) [Attene et al. 2006] is not able to identify negative volumes, making associated features unrecoverable, (2) the same method does not distinguish between the inner components versus the outer shell, producing in our case rudimentary components, and (3) resulting segmentations in the form of basic primitives remain geometrically disparate, without forming a single, abstract shell geometry. When a set of input models is considered, our approach to co-abstraction relies on the ability to relate abstractions computed from different models, requiring inter-collection metrics and analyses. Previous works [Huang et al. 2011; Sidi et al. 2011; Golovinskiy and Funkhouser 2009; Xu et al. 2010] have also used notions based on inter-collection metrics that exploit shape similarities for multi-model coherent segmentation. Our approach in particular, however, focuses on measuring *complexity* using volumetric information density as well as shape similarity, through the set of automatically synthesized abstraction geometries. The term *co-abstraction* introduced in our paper shares similar aspirations to *co-segmentation* [Sidi et al. 2011] and *co-analysis* [Xu et al. 2010], focusing on the joint processing of multiple shapes.

Applications in 3D Content Visualization. Tools for 3D content visualization involving large databases such as interactive maps are becoming increasingly more popular [Glander and Dollner 2009; Grabler et al. 2008]. In these works, a consistent abstraction of existing 3D content with identity preservation is an open challenge. For such environments, our abstraction approach can be applied directly to 3D content, also decreasing the memory footprint of the content considerably.

3 Hierarchical Abstraction

3.1 Abstraction Spectrum

Most concisely, co-abstraction involves the creation of a spectrum of abstractions (as a hierarchy) for a given model, and doing this for all the models in a collection. This ability leads to a quantitative inter- and intra-spectrum analysis such that a collection of models can be concurrently abstracted, while the resulting abstractions are both visually consistent and identity preserving. This is at the heart of our method. In light of this goal, two core challenges must be addressed: (1) A spectrum of a model must comprise individual abstractions, where each abstraction differs from others by perceptually salient geometric components (Figure 6). (2) The differences between these abstractions must be quantifiable in order to situate an abstraction relative to others within the spectrum. The volumetric constructions, which we call *subvolumes* (Section 3.2), enable us to effectively solve both of these problems. Our algorithm en-

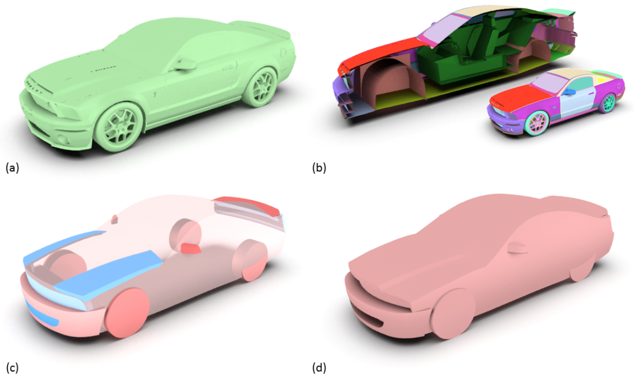

Figure 7: *(a,b) The original model consists of many isolated components, together with components inside the volume. (c) A snapshot of our abstraction process with the current abstraction in light red. Positive (dark red) and negative (blue) subvolumes are identified in multiple consecutive steps. (d) The resulting abstraction when these subvolumes are applied.*

sures that each abstraction in the spectrum differs from the others by one or more subvolumes, and that this difference is quantifiable. Finally, based on this hierarchical abstraction, we formulate co-abstraction as an entropy-minimization problem, as detailed in Section 4.

Note that previous approaches are not readily suitable for this task. For instance, one might think that the different level-of-detail abstractions computed by De Goes *et al.* [2011] or Mehra *et al.* [2009] can be used to form a hierarchy of abstractions. In both works, increases in prescribed detail levels will likely result in increased complexity in the computed curve network. However, there is no external control over whether this increased complexity introduces visually significant new components to the abstraction. Moreover, these approaches lack a basis to quantify the differences between individual abstractions, thus making a joint abstraction of multiple models difficult to compute algorithmically.

3.2 Subvolume

Man-made objects are often geometrically complex, composed of numerous details at different size scales. Different components may be separately designed, manufactured, as this approach facilitates their engineering [Ulrich and Eppinger 2011; Dieter and Schmidt 2008] (Figure 7a). Such characteristics often lead to shapes formed by a set of volumetric regions, whose unions produce the object surfaces. The volume-based view of objects is also common in aesthetic form design, where product conceptualization begins with volumetric elements such as scaffolds or inside/outside spaces ([Schmidt et al. 2009; Eissen and Steur 2007]).

Motivated by this observation, our abstraction algorithm is based on volumetric decomposition. Starting from the simplest representation of the original model, our method hierarchically identifies and applies key *subvolumes*, from which a spectrum of abstractions is produced. In contrast to methods that focus solely on surface extraction [Schnabel et al. 2007], our method can work from points condensed onto surfaces, or distributed inside a volume. In both cases, our approach seeks to identify an outer shell of the object as it would be perceived by an outside observer. By design, our method produces abstractions that vary in their visual complexity to an outside observer, while remaining oblivious to the functional and contextual intricacies. Hence, our algorithm filters out parts that may be functionally significant but geometrically minor (*e.g.* wheel bolts or the door handles of the car in Figure 7a), and parts that are significant in scale but remain inside of the shell of the object (*e.g.* the steering wheel and seats in the car).

We use the concept of subvolumes to represent volumetric information at different abstraction scales (Figure 7c). A subvolume is a bounded space that emerges from a basic volumetric primitive. After a basic primitive is fit, each face of the primitive undergoes a polynomial beautification. Starting from the coarsest abstraction level (*i.e.* model represented with the minimum number of subvolumes), our algorithm hierarchically identifies subsequent subvolumes that progressively refine the initial abstraction. A key to this process is the treatment of a subvolume as either a positive or negative volume, similar to the union and difference operations in conventional CSG. In preparation for the abstraction process, we first extract global symmetry planes, and global rotational symmetries [Mitra et al. 2006], if exists, and establish these constraints to reduce computation.

3.3 Subvolume Extraction

3.3.1 Primitive Fitting and Beautification

Basic Primitive Fit. Identifying a subvolume begins with a basic primitive fit, given the input model (Figure 8a) and the current abstraction geometry in the spectrum (Figure 8b). Our primitive fitting is based on robust statistics like those used in RANSAC methods [Schnabel et al. 2007; Schnabel et al. 2009; Li et al. 2011], where candidates are generated and scored. The main differences of our method are the utilization of volumetric primitives (Figure 5c) compared to surface primitives (Figure 5a), and our scoring mechanism.

During the abstraction process, every vertex in the original model maintains a parent subvolume it belongs to (initially, the bounding box of the original model). For each vertex, this association defines a sphere of influence with radius $R_i = \gamma D_p$, where D_p is the bounding box diagonal of the subvolume, and γ[1] is a user-defined scale ratio (Figure 8c). The current abstraction geometry consists of all the beautified and possibly blended subvolumes extracted until the current iteration (Figure 8b). The surfaces of the abstraction geometry are then segmented into clusters of *matched* or *unmatched* subregions (Figure 8e). A region on the abstraction geometry is classified as matched if it falls inside one or more of the influence spheres, and unmatched otherwise.

To assign normalized weights to the model vertices on the subsequent subvolume extraction, we calculate an impact factor to each vertex for the original model, $\mathcal{M}$:

$$P_i = A_i T_i / \sum_{j \in \mathcal{M}} A_j T_j \tag{1}$$

where P_i is the impact factor associated with the i^{th} vertex and A_i is the corresponding Voronoi area (Figure 8d). $T_i \in [0, 1]$ is the diffusion weight associated with vertex i. T_i is determined as the result of a volumetric diffusion process using the matched (yellow, $T = 0$) and unmatched (red, $T = 1$) regions of the current abstraction geometry as Dirichlet boundary conditions (Figure 8e). The steady state values of T inside the bounding box volume are computed similar to that in [Takayama et al. 2010]. However we solve for the exact diffusion on a voxelated grid using a multi-grid solver [Bolz et al. 2003] (Figure 8f). T_i for each vertex of the model is then determined from the resulting diffusion map with quadratic

[1] $\gamma = 0.05$ for all results presented here.

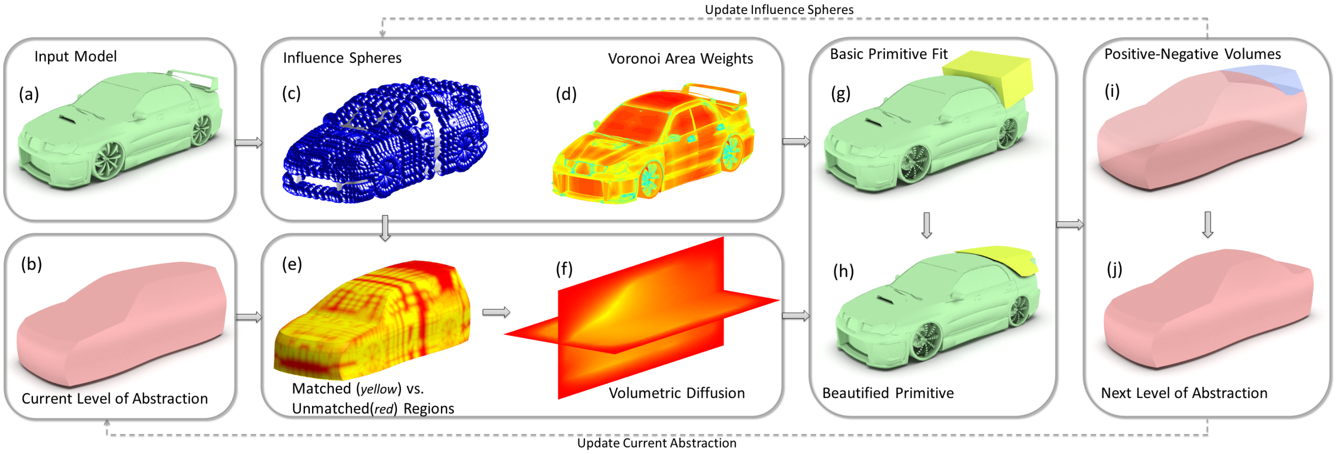

Figure 8: *(a) Input model in polygonal form possibly consisting of a number of disjoint components. (b) Current abstraction in the spectrum (coarser than the one to be calculated). (c) Influence spheres and Voronoi area weights. (e-f) Diffusion weight calculated inside the bounding box volume by means of influence spheres. (g) Basic primitive fit computed by randomized scoring. (h) Subvolume computed by beautifying the faces of the basic primitive. (i) Positivity assignment, and subsequent addition or subtraction of the computed subvolume. (j) Next level of abstraction.*

interpolation based on the 3D grid vertices. At the end, a high impact factor P_i allows the contribution of the i'th vertex to be greater in the subsequent candidate generation and scoring steps.

Our basic volumetric primitives include rectangular and triangular prisms, cylinders, and truncated cones. A number of these basic primitives are constructed using the vertices of the original model, building the candidate pool [2]. For each primitive type, the necessary number of vertices to construct it are drawn probabilistically using the impact factors in Equation 1 as the probability density function (Note that we utilize Equation 1 as a probability distribution, with A_iT_i representing the random variable. This interpretation is facilitated by the fact that: $\forall i \; 0 \leq A_iT_i \leq 1$, and $\sum A_iT_i = 1$). Rectangular prisms, for instance, require 4 vertices to be drawn. The first vertex forms one base corner. The second vertex forms the height of the prism, and together with the first vertex leads to a vector. A base plane is formed such that it passes through the first vertex and uses this vector as its normal. The third and fourth vertices are projected to this plane creating two vectors starting at the first vertex and ending at these projections. These two vectors are next orthogonalized to create the base rectangle, and therefore the full prism obtains the identified height. Triangular prisms follow the same procedure, except the base vectors are not orthogonalized. For the generation of revolved basic primitives (*i.e.* cylinder and truncated cone) we follow a method similar to that of [Schnabel et al. 2007].

In each step, the newly fit primitives are required to intersect the current abstraction geometry (non-intersecting candidates are released from the pool). This ensures that inner primitives are merged with the outer volumes when possible, and that all abstractions represent single closed volumes of arbitrary genus (hence preventing islands of solids). We assign a score, Sp, to each candidate as follows:

$$Sp = (A_{sp}/A_p) \sum_{i \in \mathcal{M}} P_i \alpha_i \tag{2}$$

where A_p is the total surface area of the basic primitive, A_{sp} is the portion of A_p that falls within the spheres of influence defined by the original vertices. The intersection area is approximated by the intersection between the sphere and an intermediate polygonal discretization of the primitive under consideration. α_i is a decay function defined within each sphere of influence as follows:

$$\alpha_i = e^{-(d_i/R_i)^2} - e^{-1} \text{ if } d_i \leq R_i, 0 \text{ otherwise} \tag{3}$$

where d_i is the minimum distance from the i^{th} vertex to the primitive surface. This scoring scheme favors primitives whose surfaces fall within the spheres of influence. Likewise, P_i draws the selection toward primitives that claim currently unclaimed volumes, and lastly α_i further favors the primitives whose surfaces are proximate to the sphere centers.

Primitive Beautification. Once the candidate with the highest score in Equation 2 is selected (Figure 8g), we beautify its faces into polynomial surfaces to match the original model (Figure 8h). For this, the vertices whose influence spheres intersect the basic primitive faces, and those that fall inside the primitive volumes are gathered. Each vertex is then assigned to the nearest primitive face forming a distinct vertex cluster Ψ_f (*e.g.* for a rectangular prism, up to six such clusters are obtained).

For a given Ψ_f, the associated primitive face is replaced with a polynomial surface. Our method attempts to identify the most suitable fit ranging from linear to quartic (or higher order, as desired) surfaces. Starting from a linear surface hypothesis, a subset of the vertices in Ψ_f are randomly drawn, and an ℓ^2-norm least-squares regression is applied. This process is repeated many times resulting in a set of linear surface candidates. Out of these candidates,

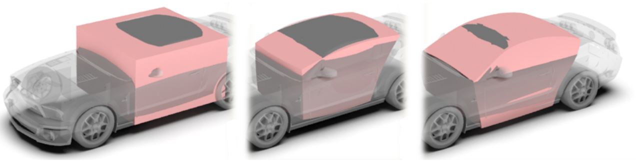

Figure 9: *(a)A basic primitive. (b)Its surfaces are beautified using conventional regression. (c)The same surfaces beautified using our approach results in a more faithful representation of the original geometry.*

[2] Candidate pool size is in 25K-100K range for all results presented here.

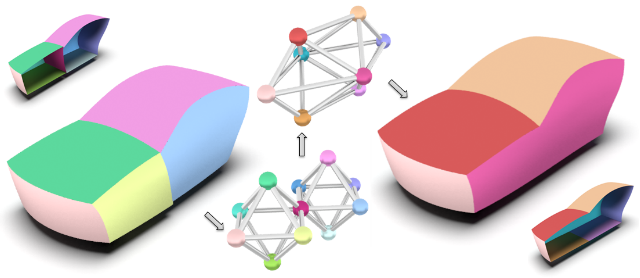

Figure 10: *Left: Subvolumes and the surface graph. Right: Surface blending and merging produce a single abstraction shell.*

the one that minimizes the following exponential-error function is chosen as the best planar surface fit to Ψ_f:

$$E^S = \sum_{v \in \Psi_f} 1 - \frac{1}{e^{d(v,S)}} \tag{4}$$

$$S^* = \underset{S}{\operatorname{argmin}} E^S$$

where E^S is the error associated with the surface hypothesis S, $d(v, S)$ is the ℓ^1-norm between v and S, and S^* is the best-fit surface.

The same process is repeated with higher degree surfaces. If the improvement in the exponential-error between the current and next-degree fit is less than 10%, the lower degree surface is kept as the beautified representation of the primitive face. Generating many surface candidates and minimizing Equation 4 allow the resulting surface to desirably snap to a subset of the vertices in Ψ_f. In effect, the selection is biased toward surfaces closely approximating a subset of the vertices, rather than a potentially looser approximation of the entire set. Figure 9 illustrates the idea. Note that the top surface of the rectangular prism claims the vertices in the original model belonging to the windshield, the roof top, and parts of the seat headrests which are inside the cabin (not visible). In Figure 9(b), a regular cubic regression approximates the windshield, the roof and the headrests, resulting in a surface that falls significantly below the roof line while remaining above the windshield. In Figure 9(c), our beautification helps represent the windshield and the roof more faithfully, leaving the headrests unaccounted for.

3.3.2 Construction of Abstraction Geometry

After a subvolume is extracted, whether it should be additive or subtractive, is decided using two scores: A union score, $S_u = \sum_{i \in \mathcal{M}_{out}} P_i \alpha_i$, where $\mathcal{M}_{out}$ are the points of $\mathcal{M}$ that are outside the previous abstraction geometry, P_i and α_i are given by Eqns. 1 and 3 respectively. This score is high when the subvolume is in the vicinity of high impact factor vertices in $\mathcal{M}_{out}$. Similarly, the difference score, $S_d = \sum_{i \in \mathcal{M}_{in}} P_i \alpha_i$, where $\mathcal{M}_{in}$ are the points of $\mathcal{M}$ that are inside the current abstraction geometry. If $S_u \geq S_d$ the interpretation of the subvolume is additive, and is subtractive otherwise.

After the subvolume is constructed, its surfaces are added to a surface graph (Figure 10). This graph encodes the topological neighborhood information of all the beautified surfaces so far. However, the newly added surfaces remain geometrically disparate. We further seek to infuse the new surfaces into the existing abstraction using surface blending and merging. For blending, we determine if a natural continuity across candidate surfaces exists using the vertices the surfaces claim at that instance. If detected, we use a partition of unity on the implicit representations [Ohtake et al. 2005] to blend the surfaces (side surfaces in Figure 10). Surface merging is applied when the edges of two neighboring surfaces are in close proximity and thus can be reduced to one (hood-roof boundary in Figure 10). Lastly, surface regions that partially or fully remain inside the resulting abstraction are removed (the surface separating the engine and the cabin in Figure 10). With the new abstraction geometry in place, all vertices are revisited to determine those to be proximate to the current surfaces using the influence spheres. Vertices whose influence spheres do not intersect the current abstraction shell are relinquished for use by subsequent subvolumes.

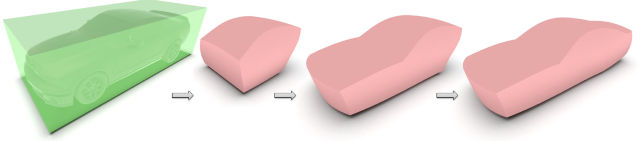

Figure 11: *Starting from the bounding box, the simplest abstraction is found in multiple subvolume extraction steps.*

The above representation has several advantages such as a small memory footprint, trivial trimmed surface calculation, rendering without tessellation, and polygonal tessellation at arbitrary resolution when needed.

Computation of the Initial Abstraction. Note that a single iteration of the process presented in sections 3.3.1 and 3.3.2 leads to the next abstraction in the spectrum given the previous coarser abstraction. However, for the initial abstraction, we perform this iteration multiple times until at least 75% of all the original vertices have been visited at least once by a basic primitive (Figure 11). This allows the starting abstraction to be a meaningful representation of the original model.

4 Co-abstraction

4.1 Method

Given a collection of models, we compute an abstraction spectrum for each model (Section 3). The goal of co-abstraction is to identify the appropriate abstraction level for each model such that all models in the collection are abstracted out to the highest extent possible, while preserving their shape identities as well as the perceived level of detail in the collection as a whole. We formulate this as an entropy minimization problem of a model collection, $\mathcal{C}$, where the total entropy of the collection (H^t) consists of (1) individual (H^n), (2) relative (H^r) and (3) mutual (H^m) entropies:

$$H^t = w_n \cdot \frac{1}{k} \sum_{i \in \mathcal{C}} H_i^n + w_r \cdot \frac{1}{k^2/2} \sum_{i,j \in \mathcal{C}} H_{ij}^r + w_m \cdot \frac{1}{k^2/2} \sum_{i,j \in \mathcal{C}} H_{ij}^m \tag{5}$$

where $k = size(\mathcal{C})$ and $i \neq j$.

Individual Entropy (H^n)**.** For a given model, the individual entropy is minimum for the most abstract form, and increases monotonically until reaching a maximum for the original geometry. Starting from the most abstract form, this term measures the added detail in each level, in relation to the volume that is added or subtracted. To this end, all models in $\mathcal{C}$ are first uniformly scaled such that each has a bounding box volume of one. With each abstraction level, the individual entropy for that particular model increases by $-\log|V_{bp}|$, where V_{bp} is the normalized volume of the subvolume added or subtracted in that level. Note that with this formulation,

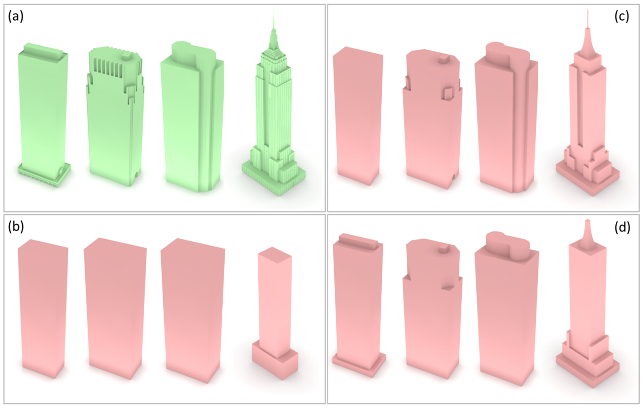

Figure 12: *Co-abstraction with different entropy terms. (a) Original models. (b) Results when only the individual entropy for each model is used. (c) Results when only relative entropies are used. (Note the inconsistent detail levels.) (d) Results when all three entropy terms are used.*

small details have a tendency to increase the entropy significantly. The individual entropy of a model at abstraction level a is then defined as:

$$H^n_a = H^n_{a-1} - \log|V_{bp}| \tag{6}$$

For models exhibiting symmetry, each abstraction level may introduce more than one subvolume. In such cases, the entropy contribution at that level is computed as the sum of the subvolume group's entropy.

Relative Entropy (H^r). Relative entropy works to keep the abstracted shapes geometrically as distinct as possible so as to preserve their identities. In the context of shape similarity, this translates into positioning each object farthest from all other objects in a shape signature space. To measure shape similarity, we utilize the two-dimensional angle-distance (AD) histogram [Ohbuchi et al. 2005], which is calculated from the pairwise euclidean distances and the inner products of the surface normals at randomly selected points over the model. To generate the AD histogram, we use 2048^2 randomly generated points on the model and map the results to a $128(distance) \times 32(angle)$ quantization of the histogram. The relative entropy between abstractions a_i and a_j, with shape histograms h^{a_i} and h^{a_j} respectively, is then defined as:

$$H^r_{a_i,a_j} = \frac{1}{\sum_{x=1}^{128}\sqrt{\sum_{y=1}^{32}(h^{a_i}(x,y) - h^{a_j}(x,y))^2}} \tag{7}$$

We also experimented with pairwise distance ($D2$) histogram [Osada et al. 2002], and the shape diameter (SD) histogram [Shapira et al. 2008]. However AD has provided the best compromise between accuracy and computation time. Note that our co-abstraction framework can be similarly used with other shape signatures [Tangelder and Veltkamp 2008].

Mutual Entropy (H^m). While relative entropy preserves geometric differences, mutual entropy aims to maintain a consistent abstraction level for the models in $\mathcal{C}$. As such, for a pair of models, it works to make their abstraction levels similar, regardless of whether the models are highly abstracted or detailed. This helps establish a perceived uniformity for co-abstraction in which the resulting abstractions all seem to share a similar level of detail (Figure 1). For two abstractions a_i and a_j, with individual entropies $H^n_{a_i}$ and $H^n_{a_j}$ respectively, the mutual entropy is then defined as:

$$H^m_{a_i,a_j} = \|H^n_{a_i} - H^n_{a_j}\| \tag{8}$$

Genetic Optimization and Global Normalization. After the abstraction spectrum of each model in $\mathcal{C}$ is calculated, the individual entropy, H^n, and the shape signature, AD, can be calculated for each abstraction of each of the models. The goal in this step is to determine the optimum abstraction for each model such that Equation 5 is minimized. We use genetic optimization where a solution vector $\mathbf{O} = q_1, .., q_k$ is defined with q_i representing a real number in the range $[0.0, 1.0]$, and k is the number of models in the collection. The fitness of such a solution is computed in three steps: (1) Transform each q_i ($1 \leq i \leq k$) to an integer between $[1, .., N_i]$, where N_i is the number of distinct models computed in the abstraction spectra of the i^{th} model. For this, we use a linear mapping function that maps the range $[0.0, 1.0]$ to $[1, ..., N_i]$, (2) Use the entries of the new integer vector as the current abstraction levels for the models in the collection, (3) Compute the total entropy, hence the fitness of the solution vector $\mathbf{O}$ using Equation 5. Starting from an initial population of 60 randomly generated such solution vectors (hence individuals), we use genetic optimization to minimize Equation 5. In each generation, we maintain a population of 60 individuals with 2-individual elitism, and apply a one-point crossover with 5% mutation rate. Note that crossover and mutation operations are applied rather straightforwardly, thanks to the common $[0.0, 1.0]$ range vectors shared by all potential solutions. Prior to optimization, we perform a range-normalization for each of the three entropy terms so as to make the contributions of the individual entropies congruent. The weights w_n, w_r, w_m in the objective function can be chosen separately for biasing when necessary (our current implementation uses equal weights).

Effect of Entropy Terms. Figure 12 shows input models co-abstracted using different entropy terms. Figure 12b shows the results when only individual entropy terms are used in Equation 5. As shown, this expectedly produces the most abstract form for each model. Figure 12c are the results when only relative entropies are used. This results in some models remaining too detailed compared to other models, which adversely affects the abstraction level consistency in the collection in an effort to maximize identity preservation. Here, the relative entropies that emphasize shape dissimilarity forces one model to its most abstract form, while the other remains more complex. Finally, Figure 12d shows the results when all three terms in Equation 5 are used, which strike a desirable balance between these individual forces.

4.2 Evaluation - Preliminary User Study

We conducted a preliminary user study to assess our method. The study focuses primarily on two key aspects of co-abstraction on a set of models: (1) Preservation of each model's identity, and (2) Mutual consistency of the resulting abstractions. Two main question types were designed and administered to evaluate each aspect as detailed below. 74 subjects with varying backgrounds participated in the study, each completing both question types. Subjects took on average 15 mins to complete the study.

Collections of the User Study. We select the collections of the user study to mimic familiar settings. In the first part of the user study, six different collections (Figure 13a-b), each composed of ten models, are used. These are: (collections 1 and 4) buildings from two city scenes, (2) chairs, seats and couches from a living room, (3) cylindrical bottle-like shapes, (5) cars from a street scene, and (6) a collection composed of various objects such as bottles, lamps, hand-held devices, coffee machines, mugs and cameras. In the second part of the user study, four different collections are utilized, each composed of ten models including the held-out model (explained in the following paragraphs). Figure 13c: buildings from a city scene with one building held-out, Figure 13d: buildings and cars with one car held-out, Figure 13e: elements from a living room

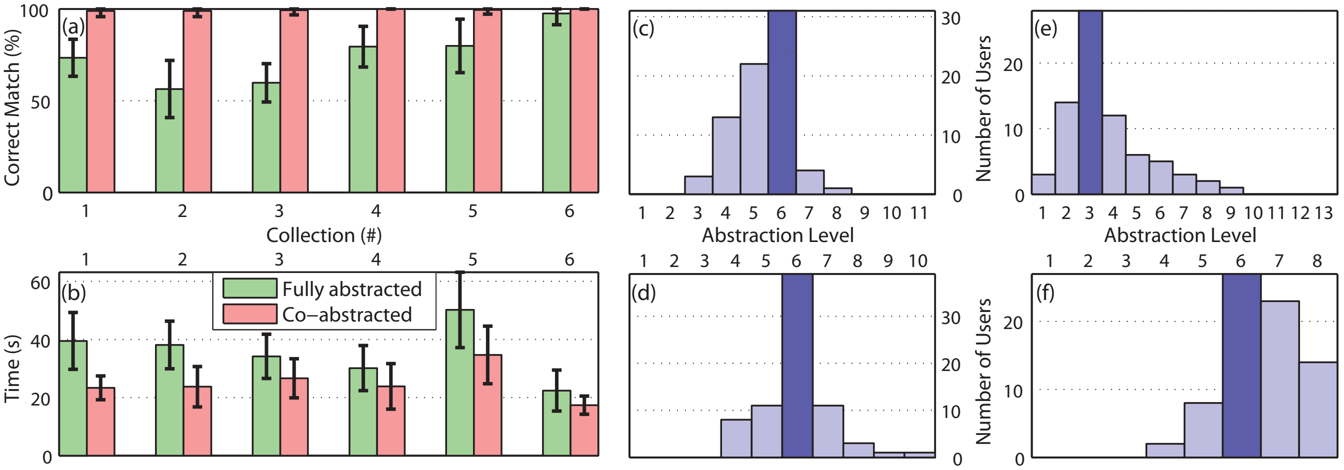

Figure 13: *User study results. (a-b) Users' matching percentage and time of completion for a model matching task: fully abstracted (green) and co-abstracted (red) versions of a model collection. Results are shown for six different collections. (c-f) Mutual consistency task: Histogram of the users selection of the best abstraction level to match the detail level in a collection of models. Results for four different problems are shown. Darker bars in the histograms show the selection of our co-abstraction algorithm.*

with a counter top held-out, and Figure 13f: models from an indoor scene with a camera held-out. [3]

Identity Preservation. In this part, the goal is to measure how well co-abstracted models preserve their identities (test), compared to fully abstracted models (control). Subjects are asked to match a given abstraction to its correct original model. In the control experiment, a fully abstracted model is presented, and the subject tries to identify the correct original model in the set. This is done for 6 different collections, where each collection consists of 10 matching problems. In the test case, the same participant matches the co-abstracted models in a similar setting resulting in 120 individual matches recorded from the participant. We randomize the order of problem presentation to eliminate a potential bias due to a consecutive presentation of the same test and control collection. The test measures the accuracy of subjects' matches, as well as the speed with which this matching is executed. The hypothesis is that fully abstracted models are significantly more difficult to identify (and thus match to the originals), compared to co-abstracted models. Figure 13(a-b) shows the matching accuracy for the six collections for the test and control experiments. For the first five sets, where the models in a set are globally similar, both the matching accuracy and the matching speed is significantly higher with our co-abstraction. In the sixth collection, where the individual models are already very distinct, full abstractions were expectedly sufficient for an accurate matching. However, fully abstracted models still took longer to match compared to co-abstracted models.

Mutual Consistency. This part evaluates whether our co-abstraction method produces results commensurate with those preferred by humans. The participants are presented with the co-abstractions of all the models in a collection, except for one model's abstraction. For this held-out model, the participants are separately presented with that model's entire abstraction spectrum. The subjects are then asked to identify the abstraction from this spectrum that is visually most consistent with the remaining co-abstracted models. Each subject's choices are recorded. For each subject, we used four different collections for this task. The hypothesis is that, for a given model, the best abstraction level determined by our method is similar to that preferred by humans. Figure 13(c-f) shows the distribution of the subjects' choices in each of the four problems. As observed, humans' choices exhibit strong consistencies, indicating the existence of a common preference among humans. Note that the optimum abstraction level computed by our co-abstraction matches this inferred preference in all four problems.

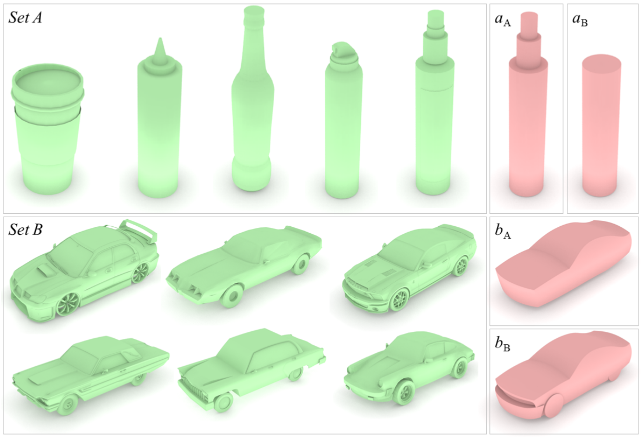

Figure 14: *Spray bottle's final abstraction when co-abstracted with Set A (a_A), and when co-abstracted with Set B (a_B). Car's final abstraction when co-abstracted with Set A (b_A), and when co-abstracted with Set B (b_B).*

5 Results and Discussions

Figure 14 demonstrates how the abstraction of a model can be influenced by the other models in the collection. When a model is co-abstracted with geometrically different models, the resulting abstraction is simpler. However, in the presence of geometrically similar objects, the abstraction maintains more details for identity preservation.

Results of our method are demonstrated with several examples throughout this paper; a city view with buildings and cars (Figure 1), a living room (Figure 15a), and objects on and around a table

[3] See the supplementary document and video for the complete set of visuals of these model collections.

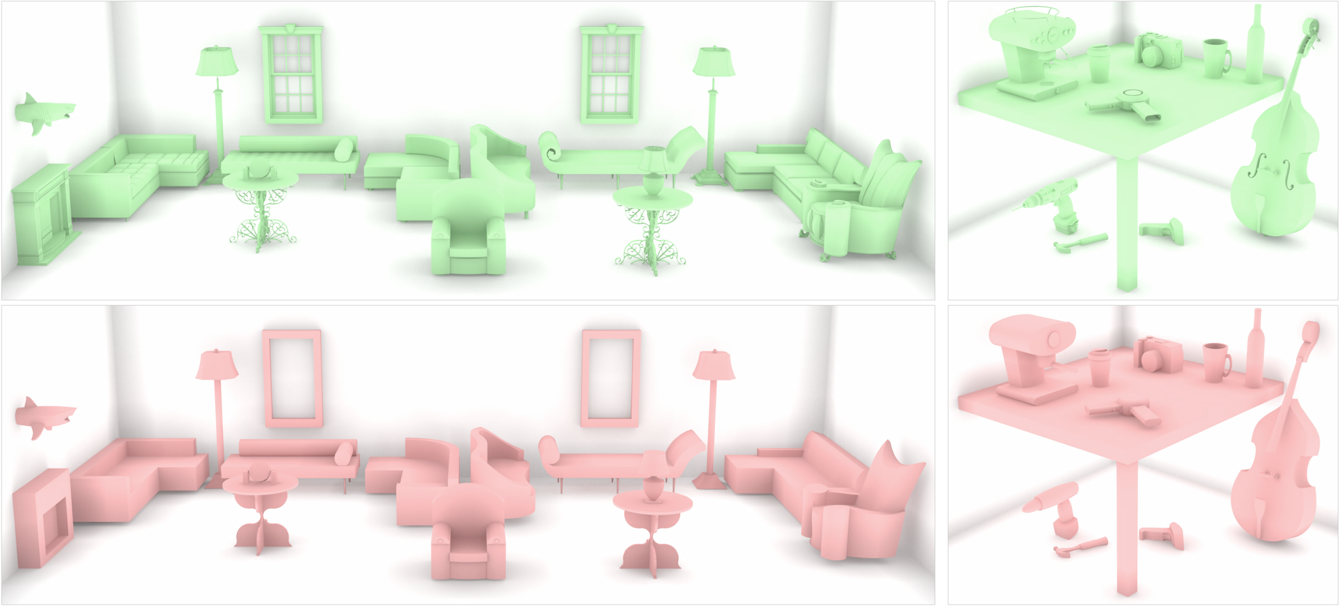

Figure 15: *Objects in a living room and their co-abstraction (left). Objects on and around a corner table and their co-abstraction (right). (Zoom in for better viewing in the digital version.)*

(Figure 15b). Our method results in abstractions where the objects in the scene are reduced to a consistent level of abstraction, while each object maintains its identity preserving attributes. As shown in our user study (Section 4.2), these outcomes are commensurate with those suggested by humans.

Figure 16 shows the abstraction results on an original and noise added model. As described previously, our method takes as input triangle soups and therefore does not require a mesh topology. Note that, the only topology dependent quantity is the Voronoi area. Since we utilize the Voronoi area as a weight per vertex (Equation 1), this area is computed considering all the faces incident on a vertex. This allows triangle soups to be handled no differently than the way a connected mesh topology is processed. As seen, while the two abstractions are generally similar, there are differences in the geometries of certain parts such as the tops of the arm rests. These differences occur primarily due to the alterations in the original geometry.

As described, our method generates a hierarchical abstraction spectrum for each model independently and then computes a joint co-abstraction afterwards. It should be noted that another possibility would be to compute both the hierarchy of abstractions and the co-abstraction concurrently so that the shape collection influences the abstraction spectrum of each shape. The consequence of this would be the creation of as many hierarchies as the number of different collections the shape appears in, which will lead to the creation of as many hierarchies for the shape as the number of collections it appears in. However, findings in human perception point to an approach similar to the one we have taken: While an object has a single physical form, different observers perceive it in different forms [Lehar 2003]. The different perceptions of the object vary by different detail levels, only enough to enable the observer to distinguish the object from others in the observer's memory [Noë et al. 2000]. Our computational model tries to mimic these findings: we capture the essence of the shape in different detail levels thereby generating a unique hierarchy for each model, and then compute a joint co-abstraction.

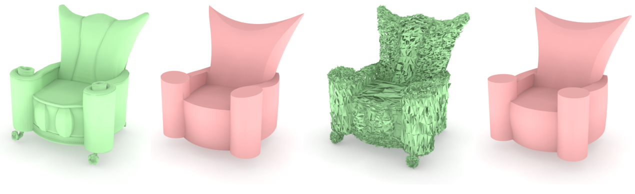

Figure 16: *Left: Original model and its abstraction. Right: Noise added model (2.5% Gaussian perturbation to vertex positions relative to the model's bounding box diagonal) and its abstraction.*

Performance. For the models considered in this article, the average time for creating an abstraction spectrum for one model is less then a minute on a 1.66GHz machine with 4GB RAM. The number of abstractions computed for these models vary from 7 to 20 as we did not restrict the maximum number of abstraction levels for any of the models. The most time is consumed during basic primitive generation and scoring, primarily because many candidates are created and evaluated, but this process suitably lends itself to parallelization, which could easily be exploited. Once the spectrum for each model in the collection is created, co-abstraction is almost instantaneous, as the necessary geometric information has already been calculated during individual abstractions. This allows model additions or removals to a collection to be easily accommodated, as new co-abstractions can be rapidly computed. Our method scales well to large collections since the processing time for hierarchy generation increases linearly with the number of models, and co-abstraction computation afterwards is much faster compared to hierarchy generation. Our abstraction scheme also allows a compact representation of the original geometry. For instance, the original chair model in Figure 16 has 29143 vertices and 56146 polygons, while the abstraction consists of 22 polynomial surfaces blended using a single blending function. Furthermore, the original models in Figure 1 are 17MB collectively in standard vertex-face list form, whereas the co-abstractions of the same are only 19KB in our implicit form. A noticeable compression ratio of $1/10^3$ of the original models.

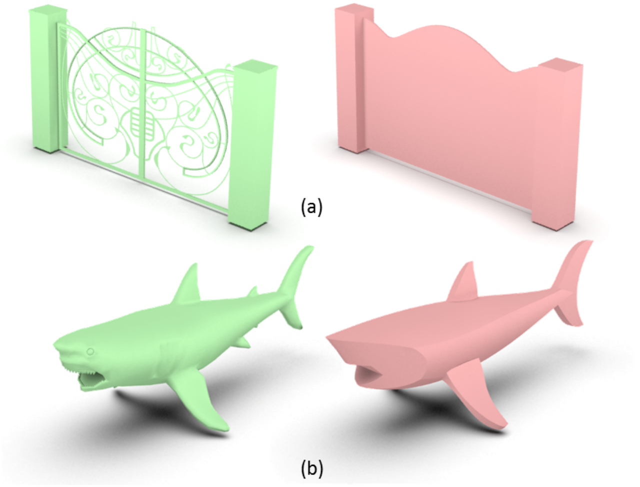

Figure 17: *Gate (a) and Shark (b) abstractions.*

Limitations. As shown in Figure 17a, if the input model contains geometric features that cannot be identified readily using volumetric components, our abstraction method washes out such details. While this may be an acceptable abstraction, it may also fail to preserve information critical in certain contexts. Also, as shown in Figure 17b, our abstraction method is not expected to perform well on organic and natural shapes. In such cases, our method may introduce surface boundaries and edges that do not exist in the original shapes.

The granularity of our abstraction scheme may also result in less than optimal co-abstraction results. The computed abstraction spectrum of a model consists of only a finite number of discrete entropy levels. In some cases this may lead to a spectrum in which an added or subtracted subvolume results in a large differences in the entropies of two consecutive abstraction models. As a result, the appropriate level of detail which is optimal for the model in question may be missed by the abstraction scheme. To alleviate this issue, future work may focus on generating a more continuous abstraction spectrum for each model. This will require the co-abstraction process to optimize the level of detail in a continuous abstraction space, as opposed to the currently utilized discrete space.

We believe this work can pave the way for future advances in efficient rendering and visual data compression. Applications in which a multitude of simplified models need to co-exist in a single context (*e.g.* virtual worlds for interactive 3D maps and games) might benefit the most. User guided coherent scene composition using models imported from different databases is also a promising direction.

6 Conclusion

In this paper, we hypothesize that there may be many admissible abstractions of a given geometry, and the search for the most suitable one only makes sense in the context of other models. To this end, we presented an identity preserving co-abstraction method to conjointly simplify the models in a collection at consistent levels. Our method uses a spectrum extraction method based on subvolumes as the main constructs for abstraction. Our approach to model abstraction is robust under imperfections common in 3D geometric data such as disconnected parts, self-intersections, and noise. Shapes co-abstracted using our method collectively maximize geometric simplicity, while each preserves the necessary details to distinguish itself from others, and result in a coherent set. Our user study has shown that the outcomes of our method are similar to those obtained from human subjects. Abstractions are represented as compact sets of polynomial surfaces, blended using simple blending functions.

Acknowledgements

We thank the anonymous reviewers for their invaluable input and suggestions. The original models used throughout the paper and the supplemental materials are copyright of the respective owners from Google 3D Warehouse, Archive-3D and Artist-3D. We thank the creators of these models for making their work publicly available.

References

Attene, M., Falcidieno, B., and Spagnuolo, M. 2006. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer 22*, 3, 181–193.

Attene, M., Mortara, M., Spagnuolo, M., and Falcidieno, B. 2008. Hierarchical convex approximation of 3d shapes for fast region selection. In *Computer Graphics Forum*, vol. 27, 1323–1332.

Bolz, J., Farmer, I., Grinspun, E., and Schröoder, P. 2003. Sparse matrix solvers on the gpu: conjugate gradients and multigrid. In *ACM Transactions on Graphics (TOG)*, vol. 22, ACM, 917–924.

De Goes, F., Goldenstein, S., Desbrun, M., and Velho, L. 2011. Exoskeleton: Curve network abstraction for 3d shapes. *Computers & Graphics 35*, 1, 112–121.

DeCarlo, D., and Santella, A. 2002. Stylization and abstraction of photographs. In *ACM Transactions on Graphics (TOG)*, vol. 21, ACM, 769–776.

Demirci, M., Shokoufandeh, A., and Dickinson, S. 2009. Skeletal shape abstraction from examples. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 31*, 5, 944–952.

Dieter, G., and Schmidt, L. 2008. *Engineering design*. McGraw-Hill Science/Engineering/Math.

Eissen, K., and Steur, R. 2007. *Sketching: drawing techniques for product designers*. Bis.

Foster, J. 2000. *The nature of perception*. Oxford University Press.

Glander, T., and Dollner, J. 2009. Abstract representations for interactive visualization of virtual 3d city models. *Computers, Environment and Urban Systems 33*, 5, 375–387.

Golovinskiy, A., and Funkhouser, T. 2009. Consistent segmentation of 3d models. *Computers & Graphics 33*, 3, 262–269.

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

Huang, Q., Koltun, V., and Guibas, L. 2011. Joint shape segmentation with linear programming. In *ACM Transactions on Graphics (TOG)*, vol. 30, 125.

Kreavoy, V., Julius, D., and Sheffer, A. 2007. Model composition from interchangeable components. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*, IEEE, 129–138.

Lehar, S. 2003. *The world in your head: A gestalt view of the mechanism of conscious experience*. Lawrence Erlbaum.

Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D., and Mitra, N. 2011. Globfit: consistently fitting primitives by discovering global relations. In *ACM Transactions on Graphics (TOG)*, vol. 30, ACM, 52.

Lien, J., Keyser, J., and Amato, N. 2006. Simultaneous shape decomposition and skeletonization. In *Proceedings of the 2006 ACM symposium on Solid and physical modeling*, ACM, 219–228.

Macrini, D., Siddiqi, K., and Dickinson, S. 2008. From skeletons to bone graphs: Medial abstraction for object recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, 1–8.

McCrae, J., Singh, K., and Mitra, N. J. 2011. Slices: A shape-proxy based on planar sections. *ACM Transactions on Graphics (TOG) 30*, 6, to appear.

Mehra, R., Zhou, Q., Long, J., Sheffer, A., Gooch, A., and Mitra, N. 2009. Abstraction of man-made shapes. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 137.

Mi, X., DeCarlo, D., and Stone, M. 2009. Abstraction of 2d shapes in terms of parts. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, ACM, 15–24.

Mitra, N., Guibas, L., and Pauly, M. 2006. Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (TOG) 25*, 3, 560–568.

Nan, L., Sharf, A., Xie, K., Wong, T., Deussen, O., Cohen-Or, D., and Chen, B. 2011. Conjoining gestalt rules for abstraction of architectural drawings. In *ACM Transactions on Graphics (TOG)*, vol. 30, ACM, 185.

Noë, A., Pessoa, L., and Thompson, E. 2000. Beyond the grand illusion: What change blindness really teaches us about vision. *Visual Cognition 7*, 1-3, 93–106.

Ohbuchi, R., Minamitani, T., and Takei, T. 2005. Shape-similarity search of 3d models by using enhanced shape functions. *International Journal of Computer Applications in Technology 23*, 2, 70–85.

Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., and Seidel, H. 2005. Multi-level partition of unity implicits. In *ACM SIGGRAPH 2005 Courses*, ACM, 173.

Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. 2002. Shape distributions. *ACM Transactions on Graphics (TOG) 21*, 4, 807–832.

Pylyshyn, Z. 2007. *Things and places: How the mind connects with the world*. The MIT Press.

Sala, P., and Dickinson, S. 2010. Contour grouping and abstraction using simple part models. *Computer Vision–ECCV 2010*, 603–616.

Schmidt, R., Khan, A., Singh, K., and Kurtenbach, G. 2009. Analytic drawing of 3d scaffolds. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 149.

Schnabel, R., Wahl, R., and Klein, R. 2007. Efficient ransac for point-cloud shape detection. In *Computer Graphics Forum*, vol. 26, Wiley Online Library, 214–226.

Schnabel, R., Degener, P., and Klein, R. 2009. Completion and reconstruction with primitive shapes. In *Computer Graphics Forum*, vol. 28, Wiley Online Library, 503–512.

Shapira, L., Shamir, A., and Cohen-Or, D. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer 24*, 4, 249–259.

Shorey, P. 1901. Aristotle's de anima. *The American Journal of Philology 22*, 2, 149–164.

Sidi, O., van Kaick, O., Kleiman, Y., Zhang, H., and Cohen-Or, D. 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Transactions on Graphics (TOG) 30*, 6, 126.

Takayama, K., Sorkine, O., Nealen, A., and Igarashi, T. 2010. Volumetric modeling with diffusion surfaces. In *ACM Transactions on Graphics (TOG)*, vol. 29, ACM, 180.

Tangelder, J., and Veltkamp, R. 2008. A survey of content based 3d shape retrieval methods. *Multimedia Tools and Applications 39*, 3, 441–471.

Ulrich, K., and Eppinger, S. 2011. *Product design and development*, vol. 2. McGraw-Hill.

Winkenbach, G., and Salesin, D. H. 1994. Computer-generated pen-and-ink illustration. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM, SIGGRAPH '94, 91–100.

Xu, K., Li, H., Zhang, H., Cohen-Or, D., Xiong, Y., and Cheng, Z. 2010. Style-content separation by anisotropic part scales. In *ACM Transactions on Graphics (TOG)*, vol. 29, ACM, 184.