可廣泛應用之三維模組與場景縮放編輯技術

黃群凱*　　陳奕麟*　　沈奕超†　　陳炳宇‡
*‡國立臺灣大學　　†中央研究院
*{chinkyell, yiling}@cmlab.csie.ntu.edu.tw　　†joshshen@citi.sinica.edu.tw
‡robin@ntu.edu.tw

ABSTRACT

本篇論文的主旨在於提供一個三維模組與三維場景的互動式縮放方法，在縮放的過程中，能有效地保存模組的結構模式並額外生成所需的三維元件，此方法能快速地提供使用者所要求的尺寸，並且廣泛應用於絕大多數的三維模組以及三維場景。本論文起先對於輸入之三維模組與場景進行重覆性模式的結構分析並進行拆解，接著利用邊界空間將屬於相同結構或相同內涵的元件進行群組化。互動式縮放方法則是利用邊界空間的變型與操作來達到三維模組的縮放。在場景縮放的過程，我們以維持場景中的三維元件的連結性與模組的空間關係做為預設條件；並提供予使用者在操作互動式縮放的過程中，加入額外的限制與含義。

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling

General Terms

Algorithms

1. INTRODUCTION

In recent years, structure-aware shape processing and editing techniques [2, 3, 5, 12] have attracted a lot of attentions in computer graphics. These techniques exploited structural properties, including *regularity* [9] and *symmetry* [1, 8], to easily adapt existing 3D shapes while preserving their structures. Although being successful, most of these techniques are designed to work in "object-level", which mainly aims to faithfully preserve and reproduce the salient features of an individual 3D object.

A 3D scene can be regarded as a collection of 3D models, which are often arranged by a set of implicitly defined rules for a multitude of purposes such as functional or aesthetic constraints. To retarget 3D scenes, "object-level" methods are not directly applicable due to the necessity of inferring the unknown dependencies between the objects in the scene. However, there is still no method which can achieve object and scene retargeting simultaneously.

To faithfully edit a 3D scene, our goal is to provide a general retargeting method that is structure-preserving in "object level" and respects the original *arrangement* or *layout* in "scene level". In this paper, we propose an interactive method suitable for retargeting both 3D objects and scenes. Initially, the input object or scene is decomposed into a collection of constituent components embraced by corresponding *control bounding volumes*. The idea of volumetric controller (e.g. axis-aligned boxes) is not new and has been exploited in 3D shape retargeting and manipulation for man-made objects [12] or architecture models [7]. Local deformations are performed within individual controllers and the influences are propagated over the whole model to achieve global deformation. Differing from previous methods, we obtain the overall retargeting through a constrained optimization by manipulating the control bounding volumes. Besides, our volumetric representation is an abstraction of space decomposition since it is used to capture not only the intra-structures (e.g. regular patterns) in the "object-level", but also semantic grouping of objects (e.g. a row of trees) in the "scene-level".

Without inferring the intricate dependencies between the components like [4, 11], we define a minimal set of constraints that maintains the spatial arrangements and connectivities between the components to regularize valid retargeting results. The default retargeting behavior can then be easily altered by additional semantic constraints imposed by users. This strategy makes the proposed method highly flexible to process a wide variety of 3D objects and scenes under an unified framework. In addition, it leads to an interactive tool that enables users to freely explore the space of valid solutions and gain fine control over the retargeting results by adding additional semantic configurations. In spite of its simplicity, the proposed method achieves more general structure-preserving pattern synthesis (e.g. *rotational* patterns lacked in the previous method [3]) in both object and scene levels.

2. OVERVIEW

The input to our system is typically a polygonal mesh $\mathcal{M} \in \mathbb{R}^3$ representing the geometry of the source *object* or *scene*. In the case of 3D scenes, $\mathcal{M}$ can be further divided into n objects $\mathcal{M}_i,\ i = 1 \ldots n$, with unknown mutual relationship. Our goal is thus to retarget $\mathcal{M}$ into $\mathcal{M}'$ in a way such that $\mathcal{M}'$ preserves the original structure of $\mathcal{M}_i$ and maintains the spatial layout between $\mathcal{M}_i$. Without loss of generality, we will hereinafter focus the discussion on scene retargeting. As will become clear soon, object retargeting is a special case in our method among which the problem is reduced to a scene containing only one object. Following the *analyze-and-edit* paradigm [5], the proposed method is composed of two main stages:

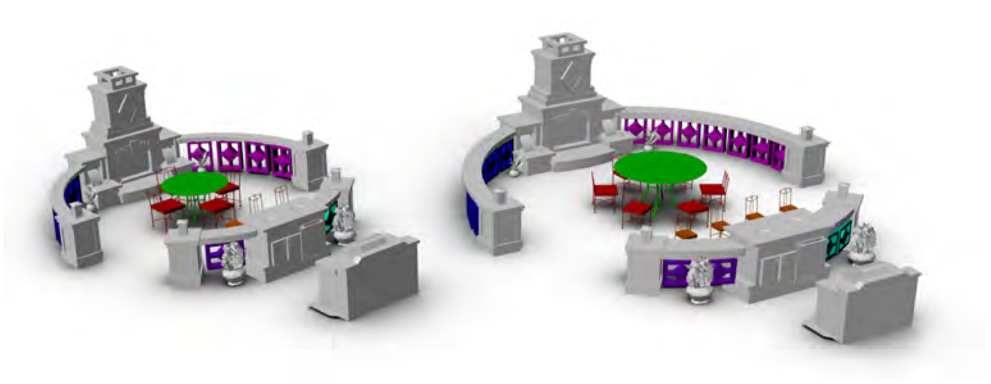


Figure 1: Given an input scene (shown in left), the proposed method generates adaptive retargeting result (shown in right) which respects the original spatial arrangement by exploiting structural regularity. The color coded parts are the detected regular patterns representing sub-shapes of objects or semantic grouping of objects in the scene.

scene analysis (Section 3) and *scene editing* (Section 4).

The goal of scene analysis is to encapsulate $\mathcal{M}$ into a set of control bounding volumes $\mathcal{V}_j$ capturing the sub-structures within $\mathcal{M}_i$ or semantic grouping of multiple $\mathcal{M}_i$. In this work, we mainly consider *structural regularity* [8, 9] presented in 3D models (Section 3.1). Once $\mathcal{V}_j$ are properly constructed (Section 3.2), we can then model valid retargeting results by solving a constrained optimization problem. To keep generality, we exploit a minimal set of constraints required to keep the retargeting results visually natural and physically valid (Section 4.1). Specifically, a set of *positional constraints* are derived from the spatial arrangement of $\mathcal{M}_i$ and are used to keep the relative order between them. Another set of *anchor constraints* accounts for the connectivity between $\mathcal{M}_i$. In the editing stage, we provide an easy-to-use system that allows users to interactively explore valid scene variations confined to the default constraints by manipulating $\mathcal{V}_j$. The default retargeting behavior can be easily modified by imposing additional constraints interactively (Section 4.2). Finally, the modified $\mathcal{V}_j$ are used to induce the overall scene synthesis results (Section 4.3).

3. SCENE DECOMPOSITION AND ANALYSIS

In this section, we explain the construction of the main auxiliary structures, i.e., control bounding volumes $\mathcal{V}$, which are used to facilitate adaptive scene retargeting. In this work, $\mathcal{V}_j$ is just a simple minimal bounding box with its main axes aligned with x-, y- and z-axis of Euclidean space, which encloses the underlying 3D shape. Unlike previous methods [7, 12], our $\mathcal{V}_j$ are not necessarily to capture physically connected geometric entities. Besides, $\mathcal{V}_j$ are also not required to be best fitted to the enclosed structures. Each $\mathcal{V}_j$ is associated with a set of parameters $(\mathbf{o}_j, w_j, h_j, d_j)$, where $\mathbf{o}_j$ is the origin of $\mathcal{V}_j$, and w_j, h_j, d_j indicate the lengths of the three main axes of $\mathcal{V}_j$, respectively. As an user manipulates $\mathcal{V}_j$, the corresponding parameters change according to the type of structure contained in $\mathcal{V}_j$ and influence the retargeting results.

3.1 Structural Regularity Detection

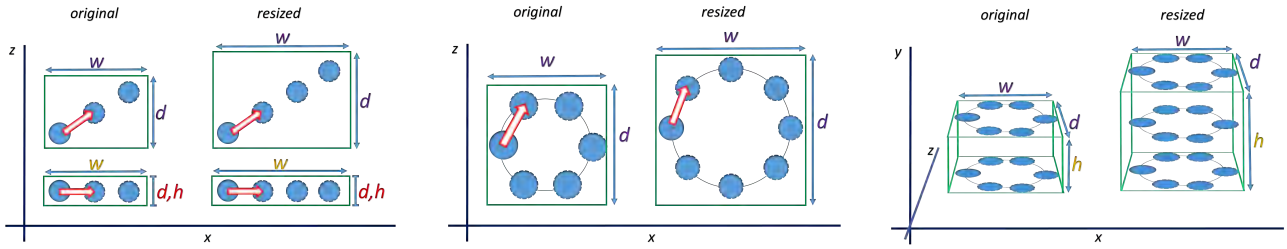
For regularity detection, we choose to apply the technique described in [9] to extract regular patterns from $\mathcal{M}$ in both object and scene levels. Specifically, regularity detection is applied to all of the $\mathcal{M}_i$ and $\mathcal{M}$ separately. By this way, not only the sub-structures among $\mathcal{M}_i$ but also the groups of $\mathcal{M}_i$ resembling regular patterns will be identified. Similar to [9], we represent each distinct regular pattern $\mathcal{P}_k$ as the following parametric form: $(\mathbf{c}_k, n_k, \mathcal{T}_k)$, which indicate the center of the starting pattern $\mathcal{P}_k^0$, the number of pattern element repetitions, and the generator transformation, respectively. We compute $\mathbf{c}_k$ as the mean position of the vertices of $\mathcal{P}_k^0$. Three types of transformation are considered, i.e., scale, rotation, and translation. According to the transformation type, the parameters of $\mathcal{T}_k$ are expressed by $(s, \mathbf{t}, \theta)$, which correspond to the scaling factor, translation vector and rotation angle. For rotational patterns, an additional parameter $\hat{\mathbf{c}}$, i.e., rotation center, needs to be derived from the pattern elements.

3.2 Control Bounding Volume Construction

Given a set of regular patterns $\mathcal{P}_k$ detected in the analysis phase, we thereby decompose $\mathcal{M}$ into a collection of constituent components embraced by corresponding bounding volumes $\mathcal{V}_i$ through the following steps:

1. For any $\mathcal{P}_k$, where $\mathcal{P}_k^0 \equiv \mathcal{M}_i$, create a bounding volume $\mathcal{V}_j$ to enclose all elements of $\mathcal{P}_k$, i.e., $\mathcal{P}_k^0 \dots \mathcal{P}_k^{nk}$. In this case, $\mathcal{P}_k$ is composed of a subset of $\{\mathcal{M}_i \,|\, i = 1 \dots n\}$, which are scattered around the scene, e.g. the chairs surrounding the tables as shown in Figure 1. Note that when $\mathcal{M}$ is a single object, there will not be this type of bounding volume.
2. For those $\mathcal{P}_k$, where $\mathcal{P}_k^0 \neq \mathcal{M}_i$ and $\mathcal{P}_k^0 \subset \mathcal{M}_i$, extract all elements of $\mathcal{P}_k$ from $\mathcal{M}_i$ and insert them into a new bounding volume $\mathcal{V}_j$. Repeat the above procedure until all $\mathcal{P}_k$ are associated with corresponding bounding volumes.
3. After all $\mathcal{P}_k$ are processed, some objects $\mathcal{M}_i$ may still have remaining parts $\mathcal{M}_i'$ which are not contained in any bounding volume. We then create a bounding volume for every separate components in $\mathcal{M}_i'$. In addition, the components in $\mathcal{M}_i'$ are conditionally split into smaller disjoint pieces in order to minimize the overlapping of the corresponding $\mathcal{V}_j$. This splitting process is beneficial because it allows a wider range of movements between physically interconnected components.

Through the above procedure, we can then obtain our control bounding volumes $\mathcal{V} = \{\mathcal{V}_j \,|\, j = 1 \dots m\}$. Note that a global bounding volume $\hat{\mathcal{V}}$ enclosing all $\mathcal{V}_j$ is also included into $\mathcal{V}$ to enable easy editing of the whole scene.



(a)

(b)

(c)

Figure 2: Illustrations of bounding volume regularization. (a) $Trans$ and $Trans*$, (b) Rot, (c) $Rot \times Trans$. The $*$ sign indicates that the translation vector $\mathbf{t}$ is not aligned with any axis of the bounding volume.

Regularity type	Size fixation	Ratio fixation
$Rigid$	w, h, d	–
$Trans$	any two of w, h, d	–
$Trans*$	any one of w, h, d	the rest of w, h, d
$Scale$	–	any two of w, h, d
Rot	any one of w, h, d	the rest of w, h, d
$Rot + Trans$	–	any two of w, h, d
$Rot + Scale$	–	–
$Rot \times Trans$	–	any two of w, h, d
$Trans \times Trans$	any one of w, h, d	–
$Trans \times Trans*$	–	any two of w, h, d

Table 1: Regularization rules of $\mathcal{V}$ for various regularity types [9]. w, h, d indicate the lengths of $\mathcal{V}$ and are interchangeable among the same row of the table.

Bounding volume regularization:. As explained earlier, we have chosen to use minimum enclosing boxes to capture various structural regularities. In spite of its simplicity, such generalization enables us to deal with a wider range of regularity types when compared with those of previous methods, such as axis-aligned [7] and translational [3] pattern synthesis. Nevertheless, when manipulating $\mathcal{V}_j$, we still need a set of regularization rules to properly define the space of valid volume sizes in order to prevent from undesired results, such as squeezing $\mathcal{V}_j$ into thin plates. It is an important feature to an interactive editing tool like the proposed method.

The main consideration of bounding volume regularization is to keep $\mathcal{V}_j$ as a minimal enclosing box after resizing, which is crucial to the semantic alignment process described in Section 4.2. Table 1 summarizes the regularization rules that we defined for different types of regular patterns and Figure 2 illustrates several cases among them. For example, as shown in Figure 2(a), when manipulating an axis-aligned translation pattern, the corresponding $\mathcal{V}_j$ is only allowed to be stretched along the direction of $\mathbf{t}$ while the other two dimensions remain constant. As for non-axis-aligned patterns, $\mathcal{V}_j$ can be stretched along the directions of the two components of $\mathbf{t}$ while the aspect ratio remains constant. For other types of regularity, such as those shown in Figure 2(b) and 2(c), please refer to Table 1 for more information.

4. INTERACTIVE SCENE RETARGETING

After the input scene is encapsulated into the control bounding volumes $\mathcal{V}$, users can then directly operate on $\mathcal{V}_j$ to obtain the desired retargeting results. To achieve this, the unknown parameters $(\mathbf{o}'_j, w'_j, h'_j, d'_j)$ associated with a modified $\mathcal{V}'_j$ need to be determined. We formulate it as a least square optimization problem with a default set of linear constraints, which assist to maintain physical connectivity and spatial layout. The space of valid scene variations can be easily altered by interactively imposing additional semantic constraints.

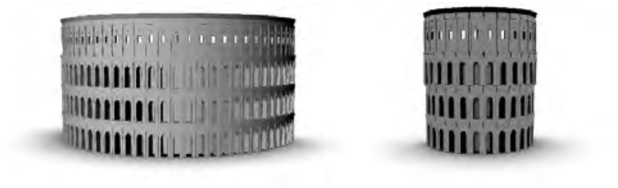


Figure 3: An example of retargeting a 3D object containing a $Rot \times Trans$ pattern.

4.1 Default Constraint Setup

Denote a point $\mathbf{p}$ lying within $\mathcal{V}_j$ as $(u, v, \omega) \in [0, 1]$, which is the local coordinate system of $\mathcal{V}_j$. We use $\mathbf{p}$ to represent a relative position in $\mathcal{V}_j$ and it can be mapped to its corresponding position $\mathbf{x}$ in global coordinate system by the following transformation function C:

$$\mathbf{x} = C_j(\mathbf{p}) = \mathbf{o}_j + (u \cdot w_j,\ v \cdot h_j,\ \omega \cdot d_j). \qquad (1)$$

For all the retargeting operations, we want to let $\mathbf{p}$ remain constant while its corresponding 3D position is updated by solving for new parameters of $\mathcal{V}'_j$.

Anchor constraints:. To obtain visually plausible retargeting results, anchor constraints are exploited to enforce physical connectivity between subdivided sub-shapes among $\mathcal{M}_i$. During the construction of $\mathcal{V}_j$, we search and record the vertices shared by different volumes when extracting or splitting sub-shapes of $\mathcal{M}_i$. Denote the set of common vertices among two bounding volumes $\mathcal{V}_1$ and $\mathcal{V}_2$ as V. For every point in V, let $\mathbf{p}_1$ and $\mathbf{p}_2$ be the points represented by its local coordinates in $\mathcal{V}_1$ and $\mathcal{V}_2$, respectively. The anchor constraint can then be expressed by the following linear equation,

$$w_a\,(C_1(\mathbf{p}_1) - C_2(\mathbf{p}_2)) = \mathbf{0}, \qquad (2)$$

where w_a is a weighting coefficient. Note that not all vertices in V are required to be included in the least square optimization. This is because the updated $\mathcal{V}'_j$ will re-compute all the remaining vertices with the new parameters which acts like applying a similarity transformation to the sub-shape enclosed by $\mathcal{V}'_j$. As a result, we sparsely select a small portion of V to form the final linear system.

Positional constraints:. The goal of positional constraints is to roughly maintain spatial arrangement between $\mathcal{M}_i$ when modifying the global volume $\hat{\mathcal{V}}$. In this work, we do not explicitly try to infer the mutual relationship between $\mathcal{M}_i$ and simply use the volume centers as reference points to place the modified $\mathcal{V}'_j$. The center of $\mathcal{V}_j$ can be expressed by $C_j(\bar{\mathbf{c}}_j)$, where $\bar{\mathbf{c}}_j = (0.5, 0.5, 0.5)$. Assume that the relative position of $C_j(\bar{\mathbf{c}}_j)$ in the global bounding volume $\hat{\mathcal{V}}$ is $\bar{\mathbf{c}}'_j$. For every $\mathcal{V}_j$, we thus impose a linear constraint as follows,

$$w_p\,(C_j(\bar{\mathbf{c}}_j) - \hat{C}(\bar{\mathbf{c}}'_j)) = \mathbf{0}, \tag{3}$$

where w_p is a weighting coefficient. Intuitively, this type of constraints encourages $\mathcal{V}_j$ to adhere to the same relative position with respect to $\hat{\mathcal{V}}$ after retargeting. Since w_p is set to a relatively small value, the original layout can still be broken and users can obtain finer control over the positioning of $\mathcal{V}_j$ by adding more interactive or semantic constraints, as explained below.

4.2 User Interaction

To understand the semantic meanings or contextual information of a 3D scene is very difficult. For example, when retargeting the scene shown in Figure 1, it might be more desirable to keep the chairs with a constant size, but is also hard to be determined automatically. It is thus our strategy to rely on moderate user inputs to provide such information and focus on designing a flexible tool to meet the common needs of 3D scene retargeting.

Semantic constraints:. In this work, we support two types of sematic constraints, i.e., *scaling* and *alignment*. By default, we allow each $\mathcal{V}_j$ to be proportionally scaled with respect to $\hat{\mathcal{V}}$. This can be modified by interactively specifying a bounding volume to freeze its size from being changed.

$$w_i\,(w'_j - w_j, h'_j - h_j, d'_j - d_j)' = \mathbf{0}. \tag{4}$$

An alignment constraint is particularly useful when we want to enforce two volumes $\mathcal{V}_1$ and $\mathcal{V}_2$ to be aligned by a plane intersecting their centers or one of the six faces. Assume the auxiliary plane l is define by a normal vector $\mathbf{n}$. In the case of center-alignment, it can be achieved by imposing the following linear constraints,

$$w_i\,(\mathbf{n} \cdot C_1(\bar{\mathbf{c}_1})) = 0, \tag{5}$$

$$w_i\,(\mathbf{n} \cdot C_2(\bar{\mathbf{c}_2})) = 0. \tag{6}$$

It is worth noting that we do not explicitly consider symmetry relationship in this work. However, the positional constraints implicitly keep symmetric relationship only when modifying the global bounding volume $\hat{\mathcal{V}}$.

Interactive constraints:. In this category of constraints, two types of user operations are supported, i.e., *displacement* and *stretching*. The displacement constraints can be regarded as a variant of positional constraint with the reference point replaced with a specific location $\mathbf{v}$ where a user would like to move $\mathcal{V}_j$. Specifically,

$$w_i\,(C_j(\bar{\mathbf{c}}_j) - \mathbf{v}) = \mathbf{0}. \tag{7}$$

It is particularly useful to adjust the position of an individual object. A stretching constraint is imposed when a user attempts to *uniformly* or *non-uniformly* enlarge or squeeze $\mathcal{V}_j$ into a new size (w, h, d). It can thus be expressed by the following equation,

$$w_i\,(w'_j - w, h'_j - h, d'_j - d)' = \mathbf{0}. \tag{8}$$

Note that when stretching a bounding volume, users can only specify a size compliant with the regularization rules listed in Table 1.

4.3 Optimization and Scene Synthesis

To obtain the global solution of the resized bounding volumes, we resort to the traditional least square minimization method to solve a linear system formed by the aforementioned linear constraints. For each regular pattern $\mathcal{P}_k$, the center of the starting element $\mathcal{P}^0_k$ is firstly updated by the local transformation of $\mathcal{V}'_j$. A new number of repetition n'_k is derived to best fit to $\mathcal{V}'_j$ and $\mathcal{T}_k$ is updated accordingly. $\mathcal{P}'^0_k$ and $\mathcal{T}'_k$ together can then be exploited to reconstruct the retargeted regular pattern. The overall retargeting result is obtained by deriving the new 3D geometry within $\mathcal{V}'_j$ according to its new parameters.

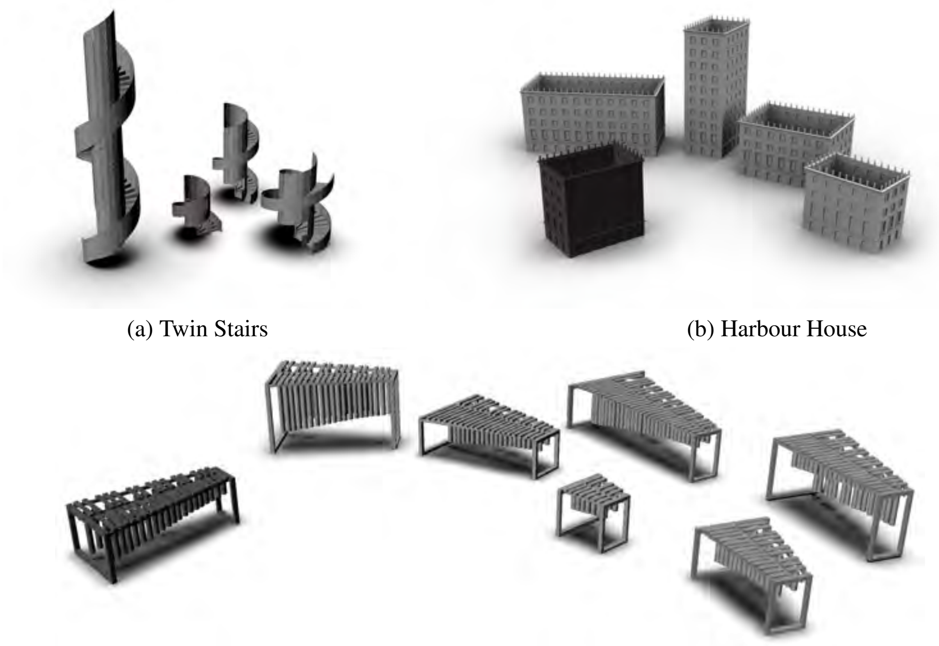
5. RESULTS AND DISCUSSIONS

We have implemented and evaluated the proposed system on a PC with an Intel i7 2GHz CPU and 8GB RAM. For solving the least square optimization, we adopted the linear solver of TACUS library [10]. Empirically, we have set the weighting coefficients for various linear constraints as $w_a = 10.0$, $w_p = 0.01$ and $w_i = 3.0$. Note that we have assigned greater influences to interactive constraints over positional constraints to enable users to alter the spatial arrangement of 3D objects in the original scene. All the test 3D objects and scenes were downloaded from Google 3D warehouse.

Performance.. The most time consuming part of our method is to perform the pattern detection [9]. It typically requires several minutes according to the complexity of the 3D objects and scenes. After this preprocessing, all the editing operations can be accomplished in less than one second, which is a reasonable response time to meet the demand of our application. Figure 5 and 6 demonstrate the effects of applying the proposed method to retarget an indoor and outdoor scenes, respectively. In these scenes, some objects scattered around form a semantic grouping resembling structural regularity (e.g. the chairs or trees). Once users modified an control bounding volume, the rest of $\mathcal{V}$ are updated simultaneously to reconstruct these patterns by inserting new elements or removing existing ones. One can see that by using the proposed system, it is very convenient to rapidly create a wide variety of scene variations that are not only visually similar but also preserve the underlying structures. Note that in Figure 6(b), the car is displaced to be aligned with the center of the house.

Comparison.. Conceptually similar to [3], we also exploit algebraic regularity and constrained optimization to model structure-preserving shape variations. However, the proposed volumetric representation is capable of modeling more general regular patterns, such as the rotational regularity presented in the Colosseum (Figure 3) or Twin Stairs (Figure 4(a)) model, which was lacked in [3]. In addition, since the pattern elements are synthesized independently of the optimization phase, our method does not suffer from the *residual bending artifacts* caused by elastic deformation [2].

Limitations.. The proposed method has a number of limitations. Firstly, for complex 3D objects without any regular patterns, our method can only produce straightforward non-uniform resizing results of the 3D object. The salient structures may not be well preserved as in [6]. Secondly, we have not yet considered global symmetries when operating on local bounding volumes. This should be able to be improved by introducing additional semantic constraints specified by users. Finally, we did not deal with the problem of collision detection in scene retargeting, which may happen when displacing or stretching a bounding volume other than $\hat{\mathcal{V}}$. We will leave it as one of the future works.



(a) Twin Stairs

(b) Harbour House

(c) Xyplophone

Figure 4: Shape variations of several 3D objects containing different regular patterns, which are generated by the proposed interactive retargeting tool. The objects in darker color are the original input models.

6. CONCLUSION

In this paper, we described a general method for interactive retargeting of 3D objects and scenes. The proposed method works on a set of bounding volumes which are capable of modeling general structural regularity. It provides an interactive and easy-to-use tool that allows users to explore a properly defined space of scene variations by solving a constrained optimization problem and is highly flexible to alter the scene configurations by imposing new constraints.

7. 致謝

本論文感謝科技部經費補助，計畫編號：NSC101-2221-E-002-200-MY2。

8. REFERENCES

[1] M. Bokeloh, A. Berner, M. Wand, H.-P. Seidel, and A. Schilling. Symmetry detection using feature lines. *Comput. Graph. Forum (Proc. of Eurographics '09)*, 28(2):697–706, 2009.

[2] M. Bokeloh, M. Wand, V. Koltun, and H.-P. Seidel. Pattern-aware shape deformation using sliding dockers. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia '11)*, 30(6):123:1–123:10, 2011.

[3] M. Bokeloh, M. Wand, H.-P. Seidel, and V. Koltun. An algebraic model for parameterized shape editing. *ACM Trans. Graph. (Proc. of SIGGRAPH '12)*, 31(4):78:1–78:10, 2012.

[4] M. Fisher, M. Savva, and P. Hanrahan. Characterizing structural relationships in scenes using graph kernels. *ACM Trans. Graph. (Proc. of SIGGRAPH '11)*, 30(4):34:1–34:12, 2011.

[5] R. Gal, O. Sorkine, N. J. Mitra, and D. Cohen-Or. iWIRES: an analyze-and-edit approach to shape manipulation. *ACM Trans. Graph. (Proc. of SIGGRAPH '09)*, 28(3):33:1–33:10, 2009.

[6] V. Kraevoy, A. Sheffer, A. Shamir, and D. Cohen-Or. Non-homogeneous resizing of complex models. *ACM Trans. Graph. (Proc. of SIGGRAPH '08)*, 27(5):111:1–111:9, 2008.

[7] J. Lin, D. Cohen-Or, H. R. Zhang, C. Liang, A. Sharf, O. Deussen, and B. Chen. Structure-preserving retargeting of irregular 3d architecture. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia '11)*, 30(6):183:1–183:10, 2011.

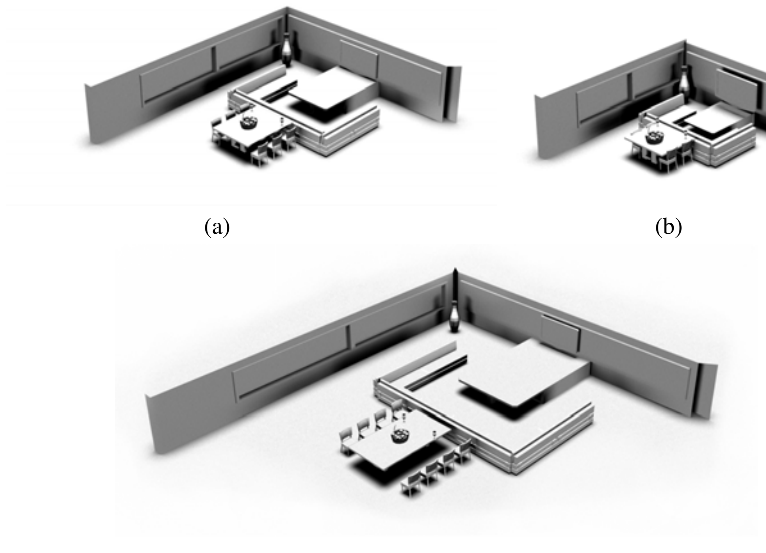
[8] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3D geometry. *ACM Trans. Graph. (Proc. of SIGGRAPH '06)*, 25(3):560–568, 2006.

[9] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas. Discovering structural regularity in 3D geometry. *ACM Trans. Graph. (Proc. of SIGGRAPH '08)*, 27(3):43:1–43:11, 2008.

[10] S. Toledo, V. Rotkin, and D. Chen. TAUCS: A library of sparse linear solvers, 2003. Version 2.2.

[11] K. Xu, R. Ma, H. Zhang, C. Zhu, A. Shamir, D. Cohen-Or, and H. Huang. Organizing heterogeneous scene collection through contextual focal points. *ACM Trans. Graph. (Proc. of SIGGRAPH '14)*, 33(4):to appear, 2014.

[12] Y. Zheng, H. Fu, D. Cohen-Or, O. K.-C. Au, and C.-L. Tai. Component-wise controllers for structure-preserving shape manipulation. *Comput. Graph. Forum (Proc. of Eurographics '11)*, 30(2):563–572, 2011.

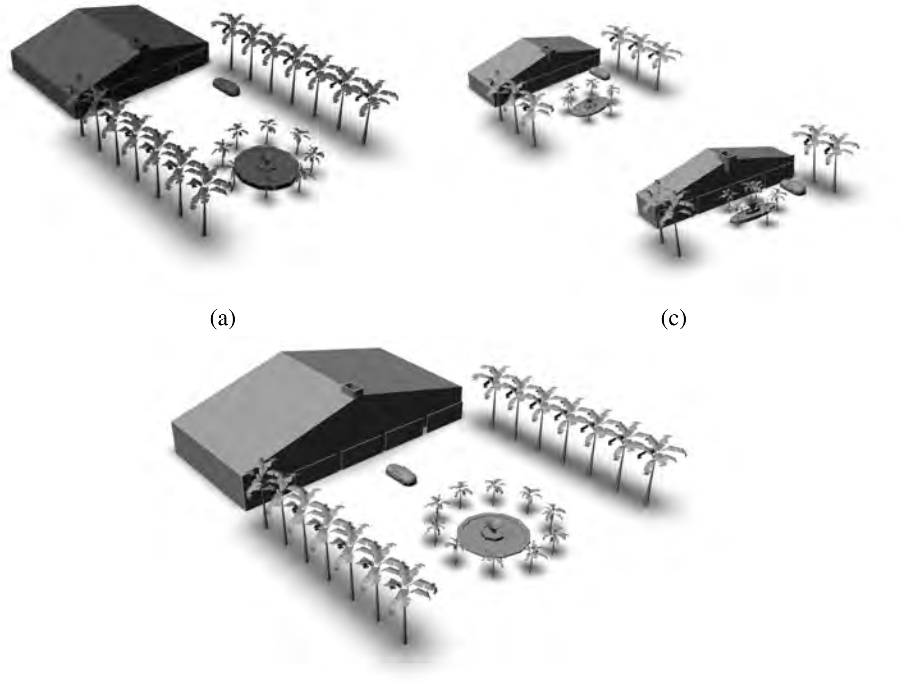


(a)

(b)

(c)

Figure 5: Retargeting of an indoor scene. (a) The original 3D scene. (b) and (c) are obtained by squeezing and stretching the input scene, respectively.



(a)

(c)

(b)

Figure 6: Retargeting of an outdoor scene. (a) The input scene. (b) The enlarged scene. (c) The scene obtained by aggressively squeezing the input scene into smaller sizes.