Two-Handed Manipulation of Tree Models

Katsuhiko Onishi Yoshifumi Kitamura Toshihiro Masaki Fumio Kishino

Graduate School of Engineering
Osaka University
2-1 Yamadaoka, Suita, Osaka, 565-0871, Japan
ohnittan@eie.eng.osaka-u.ac.jp

Abstract

This paper describes an efficient and intuitive method of tree shape manipulation by using two hands. A user can directly manipulate the shapes of tree models by using a well-designed two-handed interface. The exact manipulation command and partial shape of the user's intended tree model is automatically embodied and selected from the user's approximated indication made by using two hands. Experimental results show that our proposed two-handed method is useful for effective manipulation of tree shape models.

Key words: Virtual Environment, Two-Handed Manipulation, Tree Model, Data Description

1. Introduction

Trees have been widely used in creating various virtual environments such as cities or nature parks. In such environments, geometric models of trees have become more and more complicated. Although these complicated geometric models were previously created as simple geometric shapes with adequate textures, in recent years, such models have been created as much more complex geometric shapes by using various techniques like growth simulation. Moreover, there is increasing demand for easy and variable interactions with tree models.

Much literature has been devoted to generating realistic tree models based on many unique ideas such as a growth model [1]; however, most of them generate tree models non-intuitively by using procedural rules and/or direct inputs of numerical parameters. Therefore, it has been difficult for users to intuitively understand overall geometric shapes of the tree models and appearance when trees are located in the virtual environment. Moreover, although generation of new tree shapes by editing parts of existing tree shapes might be a solution to intuitive tree shape modeling, there are no existing systems that allow users to intuitively manipulate any part of the tree shape model with a direct manipulation interface. Since the technique of tree shape manipulation is quite different from that of ordinary CAD systems that manipulate mainly artificial objects, it requires methods that utilize the users' *kansei,* or invisible operational intentions, effectively. Therefore, in this paper, we propose an efficient and intuitive method of tree shape manipulation by using two hands.

2. Interaction with Tree Models by Using Two Hands

This section describes how users can manipulate the geometric shape of a tree model directly. It also describes the necessity of finding a method of utilizing the user's *kansei,* or invisible operational intentions.

2.1. Interaction with Tree Models

Much literature has been devoted to generating realistic tree models based on many unique ideas. The *L-studio/cpfg* [2] is one of the systems designed to generate realistic tree models. This system allows the user to generate tree models by scripts based on the L-system and Chomsky grammar. The user can edit production rules and certain parameters by using a graphical editor. The *AMAP* simulation software [3-5] is a tree modeling software based on such kinds of growth simulation. This system generates tree models using the numerical parameters defined by measuring many shapes of trees in real world. There is another system that uses the method of generating a tree model by the biological parameters such as the germination condition [6]. Lintermann and Deussen created the tree modeling system called *xfrog* [7]. That system generates tree models based on the tree model components that are assembled hierarchical in graphical user interface. That component consists of tree elements (leaves and trunk) and the type of arrangement. This allows users to change the geometric shape of tree models by changing numerical values. Tree models generated by these systems are very realistic, but the interaction methods of these systems do not use intuitive inputs such as a method that based on the direct manipulation. Another system, *ilsa* [8], can directly edit shapes of plant models that are already created. This system implements the manipulation of bending a branch by using inverse kinematics technology. However, users can only manipulate a branch. A method that generates new tree shapes by editing parts of existing tree shapes might be a solution to intuitive tree shape modeling. However, there are no existing systems that allow users to intuitively manipulate any part of the tree shape model by direct manipulation.

To provide such direct manipulation for editing geometric tree models intuitively, it is necessary to find a manipulation method that gives consideration to the characteristic feature of tree models. The characteristic feature of manipulation with tree models is that it is quite different from the method employed by conventional CAD systems, which manipulate mainly artificial objects. The method of tree manipulation depends on the user's ambiguous operational intention, that is, the user's *kansei*. Therefore, it requires methods that utilize a user's *kansei,* or invisible operational intentions, effectively. There are many kinds of user's *kansei* for editing tree models. To utilize such user's intentions effectively, it is important to change them into command forms and to recognize, for each element that constitutes those command forms, a command, an operational object, a numerical parameter, and so on.

In the real world, when people manipulate something based on his/her *kansei* or operational intention, he/she operates by using both hands. This is more efficient than using one hand. In order to manipulate the shapes of a tree model directly, two or more input channels are required. Therefore, in our method a two-hand interface is used for recognizing the user's invisible intention. In the element that constitutes the command forms, there are many kinds of operational objects, such as a group that consists of branches and leaves based on a spatial region or such a group based on a hierarchical structure. In particular, we observe an operational object and then intuitively select such an operational object group as partial shapes of tree models with both hands.

2.2. Two-Handed Manipulation

Guiard's analysis of two-handed actions [9] provides an insightful theoretical framework for classifying and understanding the role of the hands. In that literature, three principles were proposed as governing the asymmetry of human bimanual gestures. The first of these principles is the *Dominant-to-non-dominant reference.* This states that the motion of the dominant hand typically finds its spatial references in the result of the motion of the non-dominant hand. For example, when a right-handed subject writes down something on paper, he/she uses his/her left hand to hold the paper and his/her right hand to move the pen relative to the position and orientation of the paper. The second principle is the *Asymmetric scale of motion*. This states that the dominant and non-dominant hands are involved in asymmetric temporal-spatial scale of motion. The movement of the dominant hand is rapid and small-scale, while that of the non-dominant hand is slow and large-scale. For example, in handwriting, the movements of the dominant hand writing characters are more frequent and more detailed than those of the non-dominant hand adjusting the page. The third principle is the *Non-dominant hand precedence*. This states that the non-dominant hand precedes the dominant hand. In handwriting, the non-dominant hand first positions the paper, and then the dominant hand begins to write. These principles are widely known in the study of two-handed manipulation. Furthermore, the usefulness of these principles has been proven in many works. Buxton et al. [10] and Hinckley [11] have verified the characteristics of these two-hand manipulations from the perspectives of various fields.

The *Toolglass and Magic Lenses* system [12], which allows a user to operate menus or view screens with the non-dominant hand, is popular as a two-handed interaction system. However, in this system, the role of the non-dominant hand remains a mode switchover and a viewing-location alteration regardless of the characteristic feature for manipulating objects. As a modeling system by both hands in a three-dimensional virtual environment, there is also *VLEGO* [13], which proposed miscellaneous manipulation functions by a method working between both hands. In this system, when a block unit is assembled to create objects, the user can manipulate block units with both hands efficiently by restricting the degree of freedom in correspondence to the non-dominant hand. However, it is not suitable for manipulation of a partial shape with a model that has a complicated shape like a tree model.

2.3. Recognition of Tree Structure

Each of the tree models generated by modelers generally has one peculiar group structure based on the tree generation algorithm that each modeler uses. For example, there are various groups, such as the group in which two or more branches and leaves are collected and the group into which all branches and all leaves are collected. Various groups are individually defined one-by-one for every kind of tree model. If these groups are used to manipulate their tree models, a partial shape that users want to select can be chosen when the group structure has fortuitously agreed with the particular group. However, when that partial shape is not consistent with the group structure, the partial shape cannot be chosen efficiently. An example of this is when the group is defined so that all branches and leaves have a hierarchical relation with a trunk. If a user selects some branches and leaves that have a hierarchical relation with each other, they are easy to select by using the group information. However, if a user selects other branches and leaves that have a spatial relation with each other, they are not easy to select because he/she selects each branch and leaf one-by-one. This is not an efficient way. Thus, if any tree model, which has a complicated shape and is one of various kinds, has only one group structure defined by the algorithm, the user cannot efficiently

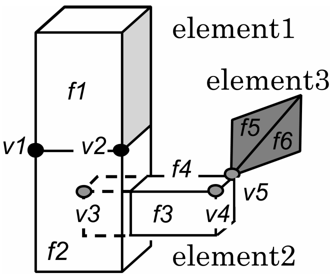


Figure 1: Division to elements.

manipulate the partial shapes of tree models with either structural selection or spatial selection. Therefore, to realize various ways to manipulate tree models, it is necessary to modify group structure dynamically in accordance with the manipulation of users by using tree-structure information, such as the connection of each branch and leaf of a tree model. For that purpose, it is necessary to reconfigure the structure of the tree model after dividing it into the smallest unit of operational object.

3. Recognition Method of Tree Structure

Here, we describe the algorithm for dynamically decomposing and reconstructing the structure of polygonal tree models. In this algorithm, each branch and leaf is classified as a minimum element for manipulation and reconstruction of those elements.

3.1. Decomposition of Elemental Groups

The tree model is decomposed based on its geometry and material data. All faces of the entire model are checked and classified into an elemental group if they share at least one vertex of the face and have the same color. For example, the faces *f1* and *f2* in Figure 1 are classified into an elemental group “element 1” because they share the vertices *v1* and *v2* and have the same color. On the other hand, the faces *f2* and *f3* are not classified into an elemental group because they do not share any vertices. The faces *f4* and *f5* are not classified into an elemental group even though they share the vertex *v5*. This is because they do not have the same color.

3.2. Recognition of Elemental Groups

All decomposed elemental groups are recognized by using the differences in geometric shape features. Here, “branch” or “leaf” for all elemental groups is determined from the difference of geometric shape features, i.e., faces consisting of a branch (leaf) form a cylinder (plane). The difference between a cylinder and a plane is based on the distribution of each normal vector. The normal vectors of a plane are distributed in a regular direction, but those of a cylinder are not. Therefore, all of the normal vectors of all faces in an elemental group are checked and classified as “branch (leaf)” if their sum is less (greater) than a threshold. In this case, the system

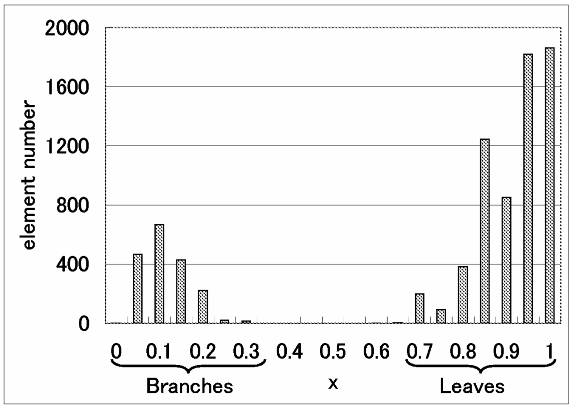


Figure 2: Distribution map of value *x*.

uses vector data at each point and classifies them as “branch (leaf).” Each element consists of *n* different vertices. Let *k* be an integer from 1 to *n* and let $\overrightarrow{vn_{(k)}}$ denote the normal vector of *k* vertex for defining the normal vector of the faces generated by that of the vertices. If the vertices were classified into one specific region, this would not be determined by only the sum of the normal vectors. This algorithm denotes *x* as the value for determining that at Equation 1. Figure 2 shows the distribution of value *x*. This distribution is of 20 tree models. It is clear that there are two ranges in Figure 2. The right one is of branch elements, and the left one is of leaf elements. Therefore, the threshold is defined as *x=0.5*. The trunk is also recognized by using its location in the object coordinate system.

The interference detection technique is used to recognize the connectivity of elemental groups. Starting from the trunk, the hierarchy of connecting branches is recognized by detecting the interference between branches. The data structure of a tree model is reconstructed as the hierarchy structure on the basis of the result from this interference detection.

$$x = \frac{\left|\sum_k \overrightarrow{vn_{(k)}}\right|}{k} \tag{1}$$

4. Selection of Tree Models by Two-Handed Manipulation

In order to select various kinds of operational groups intuitively, our method uses two widgets as a two-handed interface. It can also generate the operational group effectively as an operational object.

4.1. Dynamic Grouping of Operational Groups

There are various kinds of operational groups that are based on the user’s intention. First, the difference is dependent on the spatial position and spatial size of the region of the tree models. Moreover, even if it is in the region of the same spatial position and size, there are

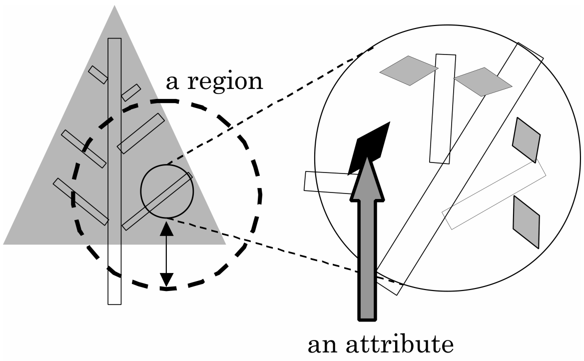


Figure 3: Selecting a region and an attribute.

different groups there to which users pay attention, such as a branch of trees or a leaf. Therefore, the difference is secondarily dependent on the attribute in the region. Accordingly, our method allows users to select the operational group by using two widgets. As Figure 3 shows, one is a region where the position and scale can be changed, and the other is the attribute to which users pay attention in the region.

Our method generates the operational group by selection of the region and attribute that the users choose based on the rule assigned to each attribute. For example, in the same region (*region 1*) that users select in Figure 4(a) and (b), if users select a leaf element as *attribute A* in Figure 4(a), our system generates the group that contains all leaves in the region. On the other hand, if users select a branch as *attribute B* in Figure 4(b), our system generates the group that contains branches and leaves that belong to a child's hierarchy from the branch selected as *attribute B*. Thus, since users select a region and an attribute in that region, our method can generate various operational groups automatically by adapting the rule of group generation for every attribute.

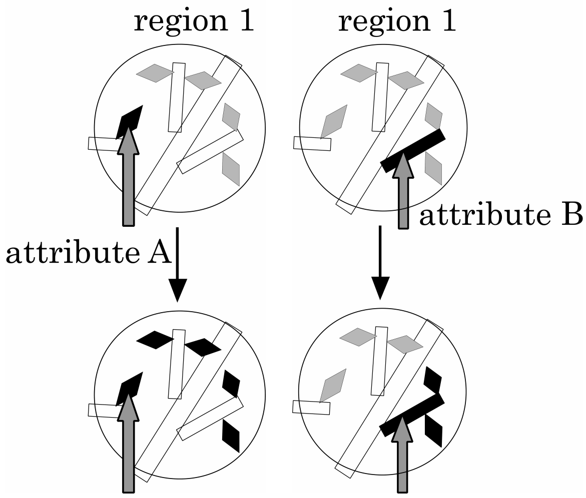
4.2. The Structure of Two-Handed Manipulation

It is necessary to assign the optional feature of the region and the attribute to each hand so that the operational group can be chosen efficiently. The theoretical framework for the asymmetrical both-hands gestures that *Guiard* proposes is that the motion of the dominant hand typically finds its spatial references in the result of the motion of the non-dominant hand. In our method, since a user selects attributes from the elements in a region, an attribute is chosen with the dominant hand and a region is chosen with the non-dominant hand.

5. Implementation and Evaluations

5.1. System Summery

In this subsection, we describe our system that utilizes a method of partial shape selection of a tree model. Figure 5(a) shows the interface of our system and Figure 5(b) shows a screenshot of our system. The cube for region selection on the presentation screen and the pointer for attribute selection are controlled by using the exclusive



(a) Attribute A. (b) Attribute B.

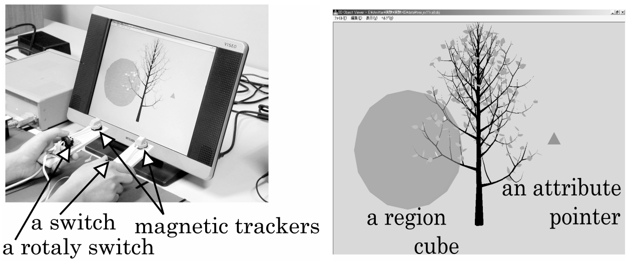
Figure 4: Example of generating operational groups.

Two-Handed interface shown in Figure 5(a). The sphere for region selection is controlled by the user's non-dominant hand, while the pointer for attribute selection is controlled by his or her dominant hand. The region and the pointer are controlled by a magnetic tracker. The region size is controlled by a rotary switch of the non-dominant side interface. The decision of the attribute is controlled by a switch of the dominant side interface. Our system is implemented on a personal computer (CPU: Pentium 3, 1 GHz x 2, Memory: 1 GBytes, Graphics board: Oxygen GVX210).

5.2. Selection of Partial Shapes by Using Dynamic Generation of Operational Groups

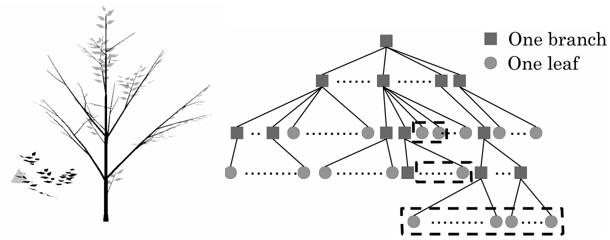
We manipulate two tree models whose tree structures are recognized. In our system, the class of a branch or a leaf is prepared as an attribute. The rule of our system that specifies group structure is dependent on the classes of the branch and leaf that are selected. That is, if users select a branch in a region, our system generates the operational group that contains all of the branches and leaves of a lower hierarchy from the selected branch. Moreover, if users select a leaf in the region, our system generates the operational group that contains all of the leaves of that region.

Figures 6 and 7 show different operational group selections for the same tree model. In each Figure, (a) shows the display of the tree shape at the time of selection, and (b) shows the tree structure at the time of selection. As shown in Figure 6(a), when all of the previously existing branches and leaves are chosen from a certain branch, the operational group is generated based on the connection relation in the tree structure, as shown by the dashed line area in Figure 6(b). Moreover, as shown in Figure 7(a), when all of the leaves contained in a certain region are chosen from a certain leaf, the operational group is generated based on the spatial



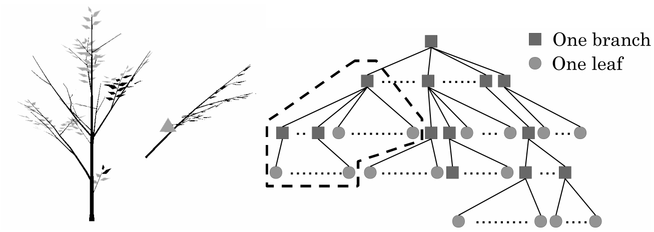
(a) Two-handed interface. (b) Screenshot.

Figure 5: Interaction system.



(a) A tree shape. (b) A tree structure.

Figure 7: Manipulation after a leaf selected.



(a) A tree shape. (b) A tree structure.

Figure 6: Manipulation after a branch is selected.

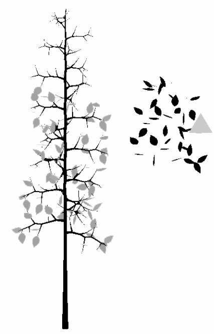


Figure 8: A tree shape. (model 2)


Figure 9: A tree shape. (model 3)

relation in the tree structure, as shown by the dashed line area in Figure 7(b). Thus, by dynamic generation of the operational group for using the selection region and attribute, it becomes simple to choose various kinds of groups for manipulation. Furthermore, as shown in Figures 8 and 9, various operational groups can be chosen similarly to the tree models of the different shapes made by the different modelers in Figures 6 and 7. That is, by performing recognition of tree structure and dynamic generation of the operational group, users can make various kinds of selections of arbitrary partial shapes according to their intentions toward the tree models.

5.3. Evaluation for Two-Handed Interface Design

We conducted experiments to validate the two-handed principles based on our interface design. We prepared two types of interface designs for this experiment. One is *the Dominant type,* which controls region size and position with the non-dominant hand and chooses the attribute with the dominant hand. This is a proposed design. The other is *the Non-Dominant type,* which controls region size and position with the dominant hand and chooses the attribute with non-dominant hand. It is the opposite function of that in our proposed design. The experimental tasks are two kinds of partial shape selections. There are three subjects who are right-handed male students. We also used two different interface designs and two different target shapes. Each subject tried to accomplish four different kinds of tasks in the experiment. As the first step of the experiment, subjects are fully trained on each interface design, and then they try to randomly complete each task ten times (a total of forty trials).

Figure 10 shows the average and variance of task completion time of each subject as the experimental results. All of the dispositions of task completion time for each subject are almost the same. However, in considering interface design, we note that for *the Dominant type*, the task completion time became shorter than *the Non-Dominant type*. Our proposed two-handed interface is the interface corresponds to the principle of two-handed manipulation, which is the *Dominant-to-non-dominant reference*, and it can be concluded that this manipulation has high working efficiency from the experimental results. Moreover, some subjects expressed the opinion after the experiment that they could move satisfactorily by using the interface of *the Dominant type* and that it is easy to use. That is, the user could intuitively manipulate objects by using the interface designed in accordance with the principles of two-handed manipulation.

6. Conclusion

We proposed a Two-Handed manipulation method for implementation in intuitive interaction with tree models having a complicated shape. In our method, we implement a selection method based on selecting a partial shape of tree models. This is evaluated through recognition of a tree model structure and dynamic generation of the operational group with both hands. Consequently, the various kinds of partial shapes of tree models could be chosen efficiently from among the various kinds of tree models. Moreover, we clarified the efficiency and intuition of our method according to the design of each function of both hands.

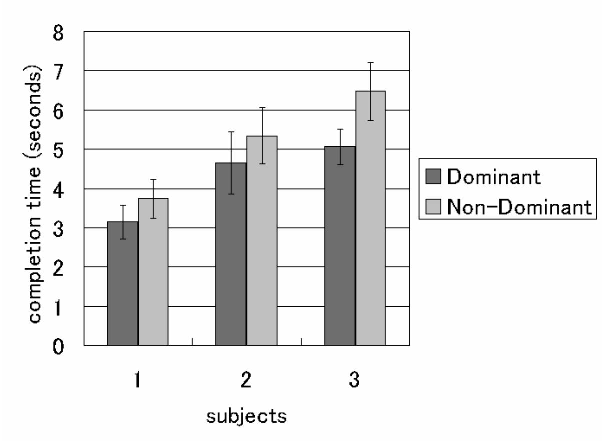


Figure 10: Task completion times.

As future study, we will pursue interactive manipulation of complicated tree models by using our method and an implementation of an editing task. We expect this to enable intuitive editing of the branches and leaves chosen by partial shape selection using Two-Handed manipulation for moving, rotation, etc., and to modify their shapes easily.

References

1. Mech, R., and Prusinkiewicz, P.: "Visual Model of Plants Interacting with Their Environment," *proceedings of ACM SIGGRAPH '96*, pp. 397-410 (1996).

2. Prusinkewicz, P., Mundermann, L., Karwowski, R. and Lane, B.: "The use of positional information in the modeling of plants," *proceedings of ACM SIGGRAPH '01*, pp. 289-301 (2001).

3. Reffye, (de) P., Fourcaud, T., Blaise, F., Barthelemy, D. and Houllier, F.: "A functional model of tree growth and tree architecture," *Silva Fenica*, Vol.31, No.3, pp. 297-311 (1997).

4. Gordin, C. and Caraglio, Y.: "A multiscale model of plant topological structures," *Journal of Theoretical Biology*, 191, pp. 1-46 (1998).

5. Gordin, C., Guedon, Y., Costes, E. and Caraglio, Y.: "Measuring and analysing plants with the AMAPmod software," Plants to ecosystem, Advances in computational life sciences, CISRO, pp. 54-84 (1997).

6. Chiba, N., Ohkawa, S., Muraoka, K. and Miura, M.: "Visual Simulation of Botanical Trees Based on Virtual Heliotropism and Dormancy Break," *The Journal of Visualization and Computer Animation*, Vol.5, No.1, pp. 3-15 (1994).

7. Lintermann, B. and Deussenm, O.: "Interactive modeling of plants," *IEEE Computer and Application*, Vol.19, No.1, pp .2-11 (1999).

8. Power, J. L., Brush, A. J. B., Prusinkiewicz, P. and Salesin, D. H.: "Interactive arrangement of Botanical L-system models," *proceedings of Symposium on Interactive 3D Graphics*, ACM, pp. 175-182 (1999).

9. Guiard, Y.: "Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a model," *Journal of Motor Behavior*, 19, pp. 486-517 (1987).

10. Leganchuk, A., Zhai, S., and Buxton, W.: "Manual and Cognitive Benefits of Two-Handed Input: An Experimental Study," *ACM Transactions on Computer-Human Interaction*, Vol.5, No.4, pp. 326-359 (1998).

11. Hinckley, K.: "Two-Handed Virtual Manipulation," *ACM Transaction on Computer-Human Interaction*, Vol.5, No.3, pp. 260-302 (1998).

12. Bier, A. E., Stone, C., Pier, K., Buxton, W., and DeRose, D. T.: "Toolglass and MagicLenses: The See Through Interface," *proceedings of ACM SIGGRAPH '93*, pp. 73-80 (1993).

13. Kiyokawa, K., Takemura, H, Katayama, Y., Iwasa, H. and Yokoya, N.: "VLEGO: A simple two-handed modeling environment based on toy blocks," *proceedings of ACM Symposium on Virtual Reality Software and Technology 1996*, pp.27-34 (1996).