3D Shapes Blending: A Development

Elleyah Goh Xin Qer
Faculty of Computing and Informatics
Multimedia University
Cyberjaya, Malaysia
elleyah.goh@gmail.com

***Abstract*—The process of initiating shapes blending begins with the input of two models of similar physical properties. These models taken have already been separated into a series of segments which allows the user to select the segment wished to be switched. Blending is then done doing the comparing of switched segment sizes and applying a rate of proportionality to detect the desired changed to be multiplied with the answer. The results are then showed onto the window with as the user desires — a partially blended model containing the chosen segment.**

Keywords—blending, shapes blending, scaling, proportionality, ratio

I. Introduction

3D modelling is the process of developing a mathematical representation of an object through the use of software, which plays a major part in contributing to computer graphics, playing frequent roles in various fields — including but not limited to animations, interior designing, architecture and so forth. With the further development of modern-day technology, the fields it has involvements in continues to grow, such as in 3D printing, which provides materials for even more fields such as dentistry. Due to the important role it commits, the development of 3D modelling has been steadily growing to ease its execution in the working life.

3D blending describes the process of applying two models or more and merging them together in order to form a completely new model based on the models used as input. By doing so, a new form is constructed as its own individual model with references to its predecessors beforehand. The application being developed for the sake of this research seeks to apply 3D blending to two various models and out of this, construct a completely new model through data processing and mathematical calculations rather than relying on a user to manually construct a merge of the two models. Hence, this aims to save time and welcomes convenience through the help of computational development.

Before going into detail, section II reviews some important literature, section III describes the methodology describes the methodology, section IV discusses the experimental results while section V contains the overall conclusion.

II. Literature Review

2.1 *'Multiple Textures Stitching and Blending on 3D Objects'* by Rocchini et al. (1999)

Based on the research done by Rocchini et al. (1999), the method used is first started from using a 3D mesh representing an object which is real. It is through this created method that the model is improved greatly with pictorial detail. In order to retrieve textural detail, the program uses a photographic process found from the real object input. The program then proceeds to register the retrieved images and stitch them onto the 3D mesh, forming a single standard texture map. Following this formation of the texture map, optimal correspondence between regions of the 3D mesh and what has been retrieved from the images is built.

It is then a new approach has been proposed by the research group for the sake of bringing these two images to map on adjacent sections of the surface based on texture blending through a smooth joint. In order for this approach to be carried out, a corresponding triangular texture patch is resampled as a weighted blend to its corresponding adjacent images sections for each mesh face connected to the adjacency border between different observed images. This accuracy of resampling and process of blending is further improved by computing an accurate local registration of the original images with respect to its current face vertices.

2.2 *'Topology Repair of Solid Models Using Skeletons'* by Zhou et al. (2007)

Zhou et al. (2007)'s research presents the development of a method used for the purpose of repairing topological errors on solid models found in the form of small surface handles, which is often an issue that arises from algorithms used for surface reconstruction.

Utilising a skeletal representation, a new mechanism is offered for identifying and measuring handles. This method provides two unique advantages compensating for previous given approaches. The first advantage given guarantees that handle removal will not introduce invalid geometrical or additional handles. The second advantage uses an adaptive grid structure, which enables the method to process large models efficiently with high resolution guaranteed.

2.3 *'Topology-Varying 3D Shape Creation via Structural Blending'* by Ibraheem et al. (2014)

Following the research of Ibraheem et al. (2014), a new algorithm is introduced with the motive of generating 3D models using topology-varying shape blending. Conclusively, this aforementioned method blends 3D models topologically and geometrically upon being given a source and target shape. As a result, the method then produces a series of merged shapes as a result of blending which can be used as new shape creations.

To start, the blending operations are firstly defined on a spatiostructural graph which has been composed of medial curves and sheets. The resulting shape abstractions have been defined as structure-oriented and part-aware, as well as it facilitates topology manipulating. Its fundamental topological operations are brought by allowing one-to-many relations between the source and target. For creative 3D modelling, an interactive, exploratory tool is used to sample and present multiple blending paths. A variety of topology-varying 3D shapes is then made via structural blending between man-made shapes containing complex real-time topological differences.

2.4 *'Shape Evolution with Structural and Topological Changes using Blending'* by DeCarlo and Metaxas (1997)

The mentioned research provides description of a framework used for gaining estimates of shape using a sparse or incomplete range of data. The shape representation used is known as blending. This allows a geometric combination of shapes to come forth into a unified model.

Estimations for the shapes using this representation is done through the usage of a physics-based framework. This includes a process which decides how to adapt both structure and topology of the model to improve the fit. This representation assists in preventing sudden changes in model geometry during fitting through allowing smooth evolution of the shape. This in turn improves the robustness of the technique.

2.5 *'Exploring Shape Variations by 3D-Model Decomposition and Part-based Recombination'* by Arjun et al. (2013)

The research uses a system which allows the creation of new shapes through the blending between shapes retrieved by a database. The shapes, prior to beginning the blending, are treated as a composition of parts. The blending process is then done by recombining the parts together from different shapes. The different shapes are selected following constraints which have been deduced by shape analysis.

The analysis perform includes shape segmentation, contact analysis and symmetry detection. This system can widely be used for instantiating new models quickly based on similar symmetry and adjacency structures according to the database though varying in appearance.

2.6 *'Contextual Part Analogies in 3D Objects'* by Shapira et al. (2009)

The primary objective of the work done by Shapira et al. (2009) addresses the issue finding analogies between parts of a 3D object. This is done via defining the similarities between two parts based on local signatures and geometry, as well as context within the shape they belong in. Through the approach designed by the team, objects are segmented hierarchically, as well as provided a local signature.

However, approaching the issue of finding corresponding parts in other objects, a context enhanced part-in-whole matching is applied to use. This method bases entirely on bi-partite graph matching and computed through the use of a flow algorithm, taking into account local geometrical features and partitioning hierarchy.

2.7 *'Consistent Segmentation of 3D Models'* by Golovinskiy and Funkhouser (2009)

Carrying out the process of segmentation onto a 3D model has been studied and researched by Golovinskiy and Funkhouser (2009) to provide an automated approach to the process, creating correspondences between them. A graph is first constructed where nodes will represent the faces upon every mesh, and edges form connections with adjacent faces within a mesh and corresponding faces within different meshes. Secondly, a segmentation is consistently developed through the clustering of the constructed graph. This allows outlier segments which are absent in every mesh.

2.8 *'3D Shapes Blending: Parts Swapping'* by Ong et al. (2019)

The mentioned research aimed to explore further into 3D blending in order to use such for providing further ideas and concepts to assist with designers or producers alike, considering manpowered work would consume much time without further assistance. The simplified method of shapes blending was applied: parts swapping, which involves a slinky-based segmentation method. The method is done following the objective to separate each semantic feature. Afterwards, the model is then separated into further parts and applied to the blending function swapping the source part to the target model.

The results generated represent a successfully blended model, though containing faulty issues such as the model possessing gaping holes. The author provided a suggestion to implement a program capable of patching the holes through matching up neighbouring vertices.

2.9 *'RealityCheck: Blending Virtual Environments with Situated Physical Reality'* by Hartmann et al. (2019)

The project named 'RealityCheck' involves the blending of reality with virtual environments by using two different methods introduced. The first method runs by first equipping eight RGB-D cameras (known as Microsoft Kinect v2). These eight cameras have been placed in positions and used as a method of rebuilding the geometric form of the world in real-time. The second method applies the usage of a different camera (Intel Realsense) and location (atop the HMD). This method is used for a reconstruction of the physical environment around the user for their perspective. The blending performed is then accomplished through a category of three different blending methods: full environment blending, salient object blending and proximity blending. All three are done for different reasons as mentioned respectively: the first for the sake of augmenting reality into the virtual environment, the second for adding environmental compositions such as furniture and the third for the user's proximity with both head and hand positions.

III. METHODOLOGY

After due research and the assessment of reports which are alike, I have successfully managed to create and come up with a design intended to carry out the work done by 3D blending in the simplest way possible. This project intends to swap property identical segments of two 3D models and make necessary adjustments onto it (e.g. enlarging the segment if it is smaller than the target's model) before swapping it in place of the target's identical segment. As an output, the first model will be displayed with segments of the second model after the swapping process has been done.

The algorithm consists a total of five used steps. These aforementioned steps are:-

• INPUT: providing the two 3D models to be used as input;

• SELECT: choosing the segments of each model to be swapped;

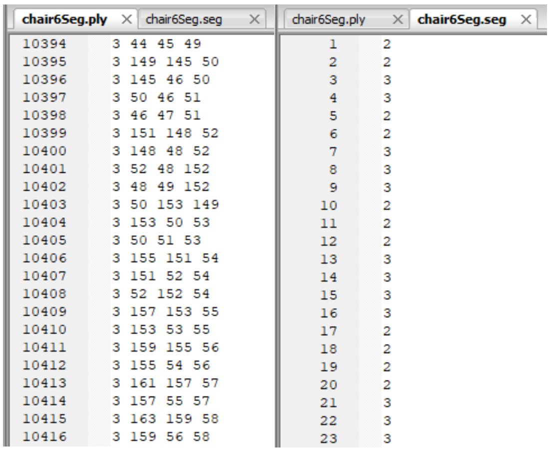
• SEPARATE: separate the chosen segments from its parent model;

• TRANSFORM: carry out transformation to morph the segment into a state appropriate for its new parent model;

• COMBINE: display the output as a result of the blending process.

3.1 Step 1: INPUT — the inputting of 3D models to be blended

When selecting models that are to be used as input (using *.ply* files), I will assume that segmentation has already been applied to the models and the files which group together parts of the segment are given as well (using *.seg* files). The latter provides an easy way of separating the parts because it is already established which nodes of the model are required to be removed without interfering with unnecessary nodes which are part of another segment.



chair6Seg.ply | chair6Seg.seg

10394	3 44 45 49
10395	3 149 145 50
10396	3 145 46 50
10397	3 50 46 51
10398	3 46 47 51
10399	3 151 148 52
10400	3 148 48 52
10401	3 52 48 152
10402	3 48 49 152
10403	3 50 153 149
10404	3 153 50 53
10405	3 50 51 53
10406	3 155 151 54
10407	3 151 52 54
10408	3 52 152 54
10409	3 157 153 55
10410	3 153 53 55
10411	3 159 155 56
10412	3 155 54 56
10413	3 161 157 57
10414	3 157 55 57
10415	3 163 159 58
10416	3 159 56 58

chair6Seg.ply | chair6Seg.seg

1	2
2	2
3	3
4	3
5	2
6	2
7	3
8	3
9	3
10	2
11	2
12	2
13	3
14	3
15	3
16	3
17	2
18	2
19	2
20	2
21	3
22	3
23	3

Figure 1: *.ply* files (left) displays the coordinates of nodes contained within a 3D model. *.seg* files (right) display which segment group the node belongs to.

Prior to running the program, these files are manually entered to the respective input variables to be used by the program. Such are found specifically in the 3D_Blending.cpp file, within the *switchModel_t(int)* and *switchModel_b(int)* void methods (the functions of these methods will be explained later).

```
void switchModel_t(int model_idx)
{
    string segment_filename = "../data/chair8Seg.seg";
    string points_filename = "../data/chair8Seg.ply";
```

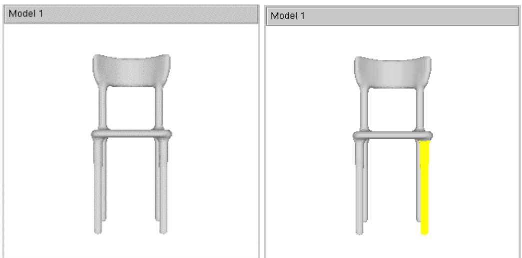
```
void switchModel_b(int model_idx)
{
    string points_filename = "../data/chair10Seg.ply";
    string segment_filename = "../data/chair10Seg.seg";
```

Figures 2 and 3: The methods containing the variables accepting the *.ply* and *.seg* files. Such files are found within the *3D_Blending.cpp* file specifically.

3.2 Step 2: SELECT — the selection of segments to be swapped

Defining targeted segments is rather a simple task to be performed prior to deciding what adjustments are to be made to them. This function is dependent on the decision of the user, which is by selecting which parts should be switched. Considering that the nodes of a segment have already been grouped together, selecting a part automatically selects with it a group of nodes that share a direct relationship with one another to form the entire segment. These segments then determine what is to occur to which part of the model.

As a method of signification, these group of nodes (the selected segment) are then shown in a different colour to inform the user as to which segment has been chosen.



Figures 4 and 5: The output shown before and after selecting the segment of the first model (the front right leg of the chair). Notice the change of colour in the targeted segment after informing the program which segment is to be swapped.

3.3 Step 3: SEPARATE — the split made between the chosen segment and its parent models

As an individual part of its own, the segment is separated from its parent model and onto the output screen for further user-friendliness. Though the change of different colour will occur to the selected segments of both input models, the output shown will vary according to whether the model is the first or the second. The reason as to why this is so is because the role of both models, despite both undergoing the process of blending, plays different acts, hence the order of input, too, easily manipulates the output as provided by the user.

The role of the first model is to act as the parent model; hence the selected segment will instead be removed from the output screen rather than simply the selected segment itself.

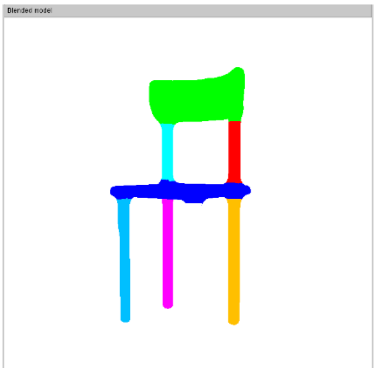


Figure 6: The output once the segment has been selected from the first model. In the case of the first model, the selected segment has been removed from the output and instead, the other parts are shown. This is because the first model is acting

as the parent model that receives a new segment to replace the selected segment it owns.

Such is not the case for the second model: the second model will display the selected segment onto the output screen itself, considering that the part selected will be swapped and will replace the segment chosen in the first model.

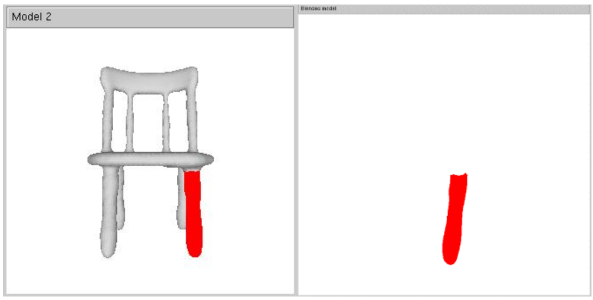


Figure 7: The selected segment of the second model and the output once the segment has been selected from the second model respectively (exclusive of the segment from the first model). This time, the output displayed is instead the selected segment rather than the parts that have not been selected.

Once both segments have been selected from both input models, the output screen will display the result of selecting before performing the action of blending. This acts as a way to make the transformation a lot more apparent before the blending has started and after it has been committed. Both displayed parts will look exactly like it did with its origin model rather than an already transformed model.

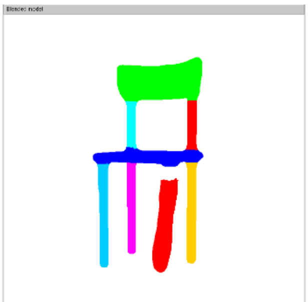


Figure 8: The output of both selected segments from both parent models. It is important to note that the blending process has not occurred yet.

3.4 Step 4: TRANSFORM — the transformation done to appropriately fit the new parent model

Transformation, the most futile step to completing the process and making the chosen parts appropriate for its new parent model, is then performed after calling upon its associated method. Mathematically, there are many methods of transformation, particularly the following:

• **Translation:** an affine transformation which moves every involved point through a certain amount as defined by the one who performs the transformation, moving through either the x-axis, y-axis and/or z-axis (in the case of a three-dimensional environment);

• **Reflection:** literally a similar action following a real-life mirror, reflecting transforms an object according to a hyperplane in the geometry into a mirrored image;

• **Rotation:** a transformation of rotating an object following a defined point and a defined angle;

• **Scaling:** a transformation upon an object which either enlarges or diminishes the target according to the scale factor.

The type of transformation covered in this program is scaling, which is applied on the selected segment of the second model once the function has been called. Before performing the appropriate scaling, the ratio between the selected segment of the first model and the selected segment of the second model is required. This method will be addressed as the usage of proportionality.

In order to do so, the initial size of both selected segments is required to conduct part-to-part comparison. The following equation is applied to first determine the ratio between both models:

$$coefficient = \frac{\max X - \min X}{\max Y - \min Y}$$

where coefficient: *coefficient of scaling*
max X: *maximum point in axis for first model*
min X: *minimum point in axis for first model*
max Y: *maximum point in axis for second model*
min Y: *minimum point in axis for second model*

After finding the coefficient signifying the ratio between both segments in its original forms, it is then applied onto the selected segment of the second model to undergo suitable transformation. The equation used is as followed below:

$$scalar = (coefficient \times \min X) + \min Y$$

where scalar: *newly scaled model*

Once the scalar value has been found, it is then implemented upon every vertex found within the selected segment of the second model. This is done to apply the usage of proportionality and have the segment possess a new form suitable enough to fit its new parent model. The blending has then been performed towards the models and are now prepared for output.

3.5 Step 5: COMBINE — the merging of segment and new parent model to form new model

With the newly transformed segment, the output is then displayed alongside the parent model from the first model as a result of the blending process. It is important to note that the unchanged model prior to executing the process and the resulting model blatantly signifies the before and after effects for it to appear more noticeable to the eyes of the user.

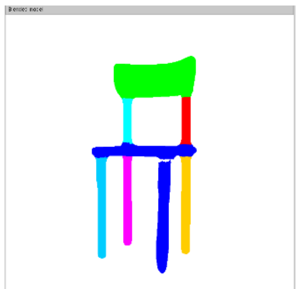


Figure 9: The output given after the shape blending.

3.6 User Interface

Three windows will be displayed in the graphical user interface: two used as input (displaying the source models) and one as an output (the final, blended model).

The user will be capable of doing SELECT by entering the segment number using the keyboard. The output shown is the first model having replaced its swapped segment with a segment from the second model.

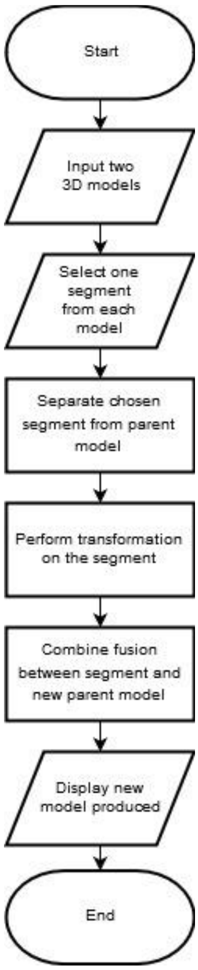


Figure 10: Flowchart of the 3D blending program.

IV. Conclusion

Viewing the results given by the program and comparing them to the objectives aimed to be achieved at the start of the program, it is safe to say that the project is a success considering that both objectives have been successfully reached:

• Objective #1: construct an algorithm that fuses part of the model into the other respective model — program has been shown to display a new model after executing the blending program, proving that it performs fusion with the source to form the target.

• Objective #2: modify the selected segment to modify its form to better suit its new model — scaling has been performed on the selected segment of the second model and applies proportionality with the selected segment of the first model to find the correct adjustments to be made.

However, even though the appointed scope of the program has been found to be met, there are still many limitations to which disrupts the performance and capabilities of it. Through working with the program and applying its usage to varying models, I have discovered that for the best performance, the selected segment from both models used require to have similar positions so the transformed segment will appear in the same location as the initial segment. Since the transformation has been said to only involve the process of scaling, the program performance plays poorly when transformations of a different kind are involved (reflection, rotation, etc). This is supported by the experimentations done, which have highly suggested that more than scaling must be applied to produce better results as an outcome. It has also been shown to lack the ability to connect the transformed segment with its new parent model, leaving unintended separation between them.

Considering the motives contributing to the completion of this project, a great deal of productivity can be brought if the program were brought to be improved. Many capabilities and potentials may be approached, particularly if we are to consider the many different kinds of transformation which can be performed. With such work done, the program will then be granted a scope larger and a limitation smaller than which contains it at the given moment. Future works are to be encouraged.

Acknowledgement

This project has been brought to work with regards and special thanks to Dr. Ng Kok Why for bringing 3D blending to my attention, as well as my close ones for motivating me to choose this path for it has piqued my interest and I felt obliged to look more into it. Without this help, this project would not have come to life and I would not have been able to discover more about the deeper structures of computer graphics as well as 3D blending.

References

[1] Alhashim, I., Li, H., Xu, K., Cao, J., Ma, R., & Zhang, H. (2014). Topology-varying 3D shape creation via structural blending. ACM Transactions on Graphics, 33(4), 1-10.

[2] Code::Blocks IDE. (n.d.). Retrieved from http://www.codeblocks.org/

[3] DeCarlo, D., & Metaxas, D. (1997). Shape Evolution with Structural and Topological Changes using Blending. Proceedings of IEEE International Conference on Computer Vision.

[4] Golovinskiy, A., & Funkhouser, T. (2009). Consistent segmentation of 3D models. Computers & Graphics, 33(3), 262–269.

[5] Hartmann, J., Holz, C., Ofek, E., & Wilson, A. D. (2019). RealityCheck. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI 19. doi: 10.1145/3290605.3300577

[6] Khronos Group. (n.d.). The Industry's Foundation for High Performance Graphics. Retrieved from https://www.opengl.org/

[7] LaValle, S. M. (2001, August 29). Homeomorphism. Retrieved from http://msl.cs.uiuc.edu/~lavalle/cs497_2001/book/cspace/node6.html

[8] Ong, K. W., Ng, K. W., & Yoong, Y.-J. W. (2019). 3D Shapes Blending: Parts Swapping, 117–121. doi: https://dl.acm.org/doi/10.1145/3372422.3372445

[9] Rocchini, C., Cignoni, P., Montani, C., & Scopigno, R. (1999). Multiple Textures Stitching and Blending on 3D Objects. Eurographics Rendering Techniques' 99, 119-130.

[10] Shapira, L., Shalom, S., Shamir, A., Cohen-Or, D., & Zhang, H. (2009). Contextual Part Analogies in 3D Objects. International Journal of Computer Vision, 89(2-3), 309–326.

[11] Zhou, Q., Ju, T., & Hu, S. (2007). Topology Repair of Solid Models Using Skeletons. IEEE Transactions on Visualization and Computer Graphics, 13(4), 675-685.