Creating 3D Human Character Mesh Prototypes from a Single Front-view Sketch

Shaikah Bakerman, Rufino R. Ansara and Chris Joslin
School of Information Technology, Carleton University, 1125 Colonel by, Ottawa, Canada

Keywords: Sketch-based Modeling, 3D Geometry, Topology.

Abstract: 3D character modeling, a vital part in film and video game production, is a process that starts by blocking out the basic geometry of a character, which is then transformed into a detailed mesh. This process can be a long and daunting task to novice modelers. To facilitate this process, we developed a sketch-based modeling system that constructs the basic 3D geometry of a human character based on a single front-view sketch and minimal user interaction. The objective of this system is to help novice 3D artists by automating the initial phase of the modeling process with an intuitive user interface, and to construct a mesh that conforms to common modeling techniques. We conducted a user study to evaluate the system's user interface and produced mesh. Results show that our interface is intuitive, easy to use and produces accurate meshes that can facilitate the modeling process. Moreover, the processing time of our system is faster than the average time it takes artists to manually construct a similar mesh.

1 INTRODUCTION

Three-dimensional (3D) modeling of human characters is a vital part of the entertainment industry. Building a 3D character commonly starts by blocking out the basic shape of the character and ends by reaching the level of detail desired (Dargie, 2007). Depending on the required level of detail, character modeling can be complex as it is constrained by professional guidelines that 3D modelers are expected to follow to ensure suitable results (Patnode, 2012).

To organize and facilitate this complex process, the common rule of building user interactive 3D modeling tools is user-centric design (Mao, Quin and Wright, 2006), where producing an intuitive and easy-to-use user interface (UI) is the main goal. However, despite the popularity and increasing use of 3D software, such as Autodesk Maya, Autodesk 3DS Max and Blender, both experts and non-experts consider these tools complex and difficult, and demand better interfaces (Cook, Agah, 2009). Moreover, per Alhashim (2015), novice 3D modelers find that using existing shapes to model a complex mesh is easier and less tedious than starting the process from scratch. This has led to the emergence of sketch-based modeling (SBM) techniques that strive to find efficient and accurate approaches to automate parts, if not all, of the modeling process.

1.1 Sketch-based Modeling

SBM is the process of automatically creating a 3D model based on one or more sketches of an object. One SBM method, called interactive sketching, is used to dynamically construct and modify a 3D model using the system's UI (Andre and Saito, 2011). Another method, consistent with our proposed methodology, involves providing images to the system for processing (Entem et al, 2015).

1.2 Research Contributions

Our main contribution is an SBM system that constructs the 3D geometry of a human character with a single image. Our goal is to facilitate the character creation process. To do this, we built a script compatible with Autodesk Maya, a common 3D modeling application. Our algorithm constructs a 3D model that conforms to the geometrical constraints of common practices, bringing the SBM concept closer to the standards of an industry that

Bakerman, S., Ansara, R. and Joslin, C.
Creating 3D Human Character Mesh Prototypes from a Single Front-view Sketch.
DOI: 10.5220/0006623702480255
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 1: GRAPP*, pages 248-255
ISBN: 978-989-758-287-5

has yet to give its full support to SBM systems (Olsen et al., 2009). In addition, we introduce an original and simple UI that dynamically displays user input on the sketch, and visually summarizes the 2D processing results to the user.

By producing simple geometry, our system strives to minimize the time it may take for modelers to fix possible artefacts. Compared to similar systems, our system focuses on the production of professional-grade geometry and topology details (Patnode, 2012), which includes the combination of body parts to produce accurate edge flow around joints.

Finally, we introduce a novel and simple approach in 2D outline smoothing, 2D skin-skeleton mapping, limb axis correction, depth estimation and mesh smoothing.

2 RELATED WORKS

2.1 Overview

Since the mid-1980s, SBM systems, including single-view (Alexe, Gaildrat and Barthe, 2004; Karpenko, Hughes and Raskar, 2002), multiple-view (Zhou et al., 2016), and data-driven systems, emerged to help 3D modelers by creating automatic sketch interpretation systems (Kazmi, You and Zhang, 2014). Teddy (Igarashi, Matsuoka and Tanaka, 2007), a single-view system, is considered the pioneer of interactive sketching UIs that focus on 3D free-form design (Kazmi, You and Zhang, 2014), although it produced only simple models and provided limited features. Other SBM systems that further contributed to simple interactive free-form modeling techniques include SmoothSketch (Karpenko and Hughes, 2006) and FiberMesh (Nealen et al., 2007).

Another important single-view system is introduced by Gonen & Akleman (2012), which constructs a "mostly" quad-based model based on curve partitioning and classification of a sketch. On the other hand, Rivers et al. (2010) propose a notable multiple-view system, which focuses on modeling man-made objects. Viana et al. (2014) propose another notable approach to create 3D models from multiple-view concept arts, which focuses on producing high quality objects at a low cost.

There are many data-driven systems, which use databases of 3D templates to construct the final 3D model (Johnston et al., 2015; Mao, Qin and Wright, 2009). The most relevant data-driven system, to our approach, is the Virtual Human Sketcher (VHS) (Mao, Qin and Wright, 2006), an established SBM system that produces simple human-shaped geometry. However, VHS differs from our approach because it relies on user interaction in each phase of the development. In addition, although the geometry is simple and flexible, the generic template restricts users to only one character style. VHS, as demonstrated by the authors, is suitable for low-quality crowd modeling but not for main characters in games or movies.

2.2 Single-view, No-database Systems

As our method is a single-view, non-data-driven system, this section covers the current state-of-the-art in this specific subcategory of SBM.

A notable system that focuses on constructing 3D characters from single-view images is by Buchanan et al. (2013). This system reads a concept image of a character and extracts a 2D skeleton from the character using an adapted smoothing and merging algorithm. For the 3D mesh, the system creates a set of arcs, where each is based on a point on the outline and a center point on the skeleton. Finally, a cross-section is extruded along the skeleton to create the 3D mesh surfaces.

The system introduced in Buchanan et al.'s work (2013) is a novel approach that produces 3D characters from single-view concept images. However, like Johnston et al.'s research (2015), this system relies on detailed and colored concept images to estimate the skeleton and character orientation more accurately. Therefore, we assume that using simple reference sketches can produce less accurate mesh results. Moreover, the 3D "arcs" revolving 180 degrees around the skeleton create a different topology than the standard topology used in game and movie characters.

Bessmeltsev et al. (2015) created a system that processes all types of cartoon characters. Their system integrates a 3D skeleton to help define the depth of the character and eliminate the ambiguities of single-view modeling. The system partitions the character outline, into body parts, based on the skeleton's joint-connected bones. The system further ensures that the symmetry of the character is maximized. It then creates 3D surfaces by revolving each surface about its 3D axis. The final 3D model aims to conform to the character outline, and create a continuity-persistent model. This system avoids the disadvantages that Buchanan et al.'s (2013) system faces, by manually producing the skeleton (Bessmeltsev et al. 2015). Moreover, it more accurately estimates the depth and the occluded parts

of the character based on the 3D skeleton. However, while the algorithm creates high-quality meshes, there are topology issues around joined areas, which the system itself does not resolve, as stated by the authors.

3 METHODOLOGY

3.1 System Overview

To model a character's basic geometry, our system automatically traces the outline of the sketch. It then displays a simple UI to allow the user to locate important body joints on the sketch. Then, using the outline and user input, it automatically derives additional joints, forming a rough skeleton and mapping between the outline and skeleton. Finally, the system estimates the body's depth, constructs and attaches body parts and creates a low-polygon count 3D character mesh. This paper details our complete process, from the initial input to the final 3D model.

We used Python 2.7, specifically PySide's QtGui and QtCore libraries to create a dynamic UI and process user input. We use the OpenMayaUI module to open the UI window in Maya, and we use .mel commands in Python to automatically develop the 3D mesh in Maya's workspace.

3.2 Sketch & Input Processing

To produce expected results, the character sketch must meet the following requirements:

1) White or very light-colored background, and clear outline of the character.
2) A human character in front view, with the arms extended 45 to 90 degrees away from the body. Fitted clothing is required.
3) Character's hair is chin-length or shorter. No loose garment, gear or other objects.

To extract the character outline, the system converts the image to binary by finding the foreground and background pixels. Next, like Buchanan et al. (2013), we use Potrace (Selinger, 2003) to determine the outline of the character sketch. Then, we apply a pixel-direction algorithm, inspired by Lin & Wang's (2011) feature detection method, to reduce pixel tracing roughness.

Our UI displays the character image with a visible traced outline (Fig. 1). Users are required to input a total of 10 points, locating the neck, biceps, wrists, hipbone and ankles. The system organizes the user-specified points in a data structure that represents a rough skeleton for the character. With point type recognition, the system only needs to distinguish between the two points of the same level (Fig. 2).

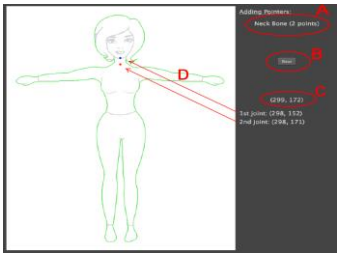

Figure 1: A screenshot of the UI.

Figure 2: The outline result (in green) of the character, with labeled Input Points (User Perspective).

Character Body Sectioning

To add skin-skeleton mappings to the arm (Fig. 3), the system divides the arm axis into *s* equally sized sections by placing *s-1* midpoints on the arm axis from the wrist to the bicep. After that, like the bicep and wrist points, the system extends two lines perpendicular to the arm axis from each midpoint, towards the outline, to find the intersection points, and then centers each midpoint. The arm axis is no longer a straight line after the adjustment of midpoints.

As the hip-leg relation is not equivalent to the bicep-arm relation, we process the legs differently. First, the system creates a leg axis from the hip to the ankle points. Then, it extends horizontal lines from the ankle point on both sides. Then, the system tries to locate the top middle point between the legs, the *crotch point*. To locate the crotch point, the system uses the inner leg curve that starts at the right intersection point of the horizontal line extended from the ankle point of the left leg and ends at the left intersection point of the horizontal line extended from the ankle point of the right leg. Following this, the system finds the crotch point, the highest pixel

on the inner leg curve, or the median of the highest pixels.

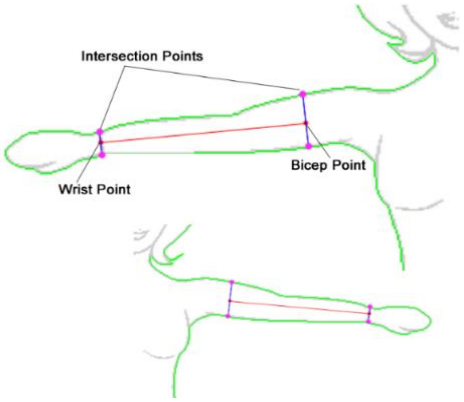

Figure 3: Skin-skeleton mapping of arm points. The system extends a line (red) from the bicep to the wrist, forming the arm axis. It then extends lines (blue) perpendicular to the arm axis, towards the outline, and looks for the intersection points (purple) with the skin. Finally, bicep and wrist points are centered between the two purple intersection points.

Using the crotch point, the system locates the pelvis corners. Pelvis corners are two points, one on each side of the crotch point, on the inner leg curve, horizontally furthest from the crotch point and lower by 1 pixel. Using those pelvis corners, the system creates the thigh line on each leg, which starts at the leg's pelvis corner and ends at the intersection point of the outer side of the leg.

When the initial limb axis fails to fit in the corresponding limb due to the limb's size or angle, our algorithm automatically adjusts the axis when it fails to find an intersection point for every perpendicular line extended from the axis midpoints. Fig. 4 demonstrates the automatic axis correction process.

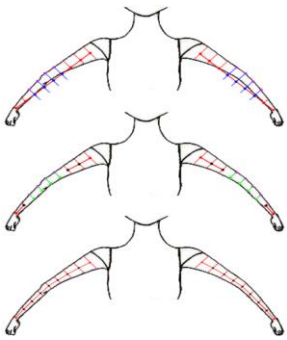

Figure 4: Limb Axis Correction Process. Top: some midpoints (blue) have intersection points (purple) from one side only because of unfitted axes. Middle: using single-sided intersections, the system extends perpendicular lines (green) towards the opposite side to find the second intersection. Bottom: midpoints are centered.

To section the neck, the system finds intersections between the two neck points and the outline. Our algorithm calculates the neck width by vertically traversing the neck to retrieve the smallest horizontal diameter in that section, forming a middle neck line with that width. After that, to determine the width and height of the head, the system applies two different approaches (see Fig. 5). The first approach calculates the head's height h_1 from the sketch and derives the width w_1, which is equal to two-thirds of h_1. The second approach calculates the head's width w_2 from the sketch and derives the height h_2 as follows:

$$h_2 = 3/2\, w_2 \quad (1)$$

The system uses the values of the first approach by default, unless w_2 is smaller than w_1. As seen in Fig. 5, h_1 is the length of l, the vertical line between the head tip and the neck top point. In addition, w_2 is the smallest horizontal diameter of the middle section of the head. Having calculated dimensions of each approach, the system uses values of the second approach if and only if w_2 is smaller than w_1. In addition, if the second approach is used, a new head tip is calculated and a new vertical line l is formed to match the new height of the head.

Finally, unlike other body parts, vertical head sectioning, using eight lines, does not depend on intersection lines but on standard head proportions that are used to approximate the widths of these lines.

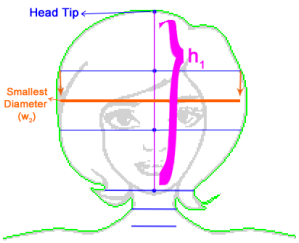

Figure 5: Head Dimensions. For 1st approach: *h1* is the length of *l*, the line between the head tip and the neck top point. For 2nd approach: *w2* is the smallest diameter found in the middle section of the head.

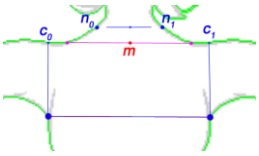

Figure 6: The 1st method of forming the clavicle line. The system searches for the lowest pixels in the outline portion (green) between c_0 & n_0, and between c_1 & n_1. The clavicle points (purple) are the lowest, and closest to m, pixels forming the clavicle line.

For the torso, the system first locates the underarm curve using an automatic body feature extraction algorithm (Lin and Yang, 2011). Then, as seen in Fig. 6, guided by the neck section and underarm curve, the system estimates the character's clavicle line. In the cases where this approach doesn't work, the system forms a clavicle line that has the same width as the neck bottom line. Next, to prepare for sufficient 3D geometry in the shoulder area, our algorithm adds a *shoulder* line, between the clavicle and chest lines, with a width equal to the average width of the two lines.

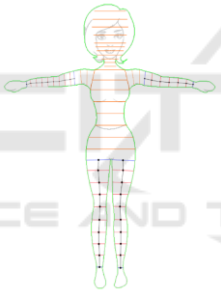

Figure 7: A complete 2D blueprint of the sketched character.

For the rest of the torso, the system inserts five equally spaced horizontal lines between the chest and thigh lines and forms the area of each shoulder with three lines placed between the upper torso and the arm. Fig. 7 presents the final 2D blueprint of all the section lines that the system automatically derives using the 10 user-input points.

3.3 3D Construction

Our 3D construction process is divided into depth estimation, mesh construction and mesh smoothing. We implemented some depth constraints that are suitable for human characters, and maintained the flexibility to create nonrealistic body proportions.

Similar to its width and height, we derive the head's depth using relatively standard proportions, which we adjusted based on empirical tests with a variety of character sketches, and it is equal to seven-eighth the height of the head. For the limbs, our algorithm simply makes the depth of each limb section line equal to its width, creating circular arms and legs. This decision is influenced by the box modeling process, where limbs are blocked out using cylinders (Villar, 2014). In addition, for simplicity, the algorithm keeps the neck middle and bottom lines circular, but locally manipulates them during the smoothing process.

To model the torso, the chest depth d_c is first calculated based on empirical tests to estimate a suitable torso depth, and is influenced by NASA's anthropometric data (Churchill et al., 1978). In addition, d_c is constrained by width w_c of the chest line. d_c is first assigned the greater value among the two following expressions:

$$d_n * 1.05 \quad (2)$$

$$d_b + d_r \quad (3)$$

where d_n, in (2), is the depth of the neck bottom. In (3), d_b and d_r are the depths of the bicep lines of the left and right arms, respectively. If d_c is greater than w_c, then it is adjusted to be:

$$d_c = {}^{w_c}/_{2} * 1.45 \quad (4)$$

Next, the system calculates the depth of the clavicle and shoulder lines based on the quadratic Bezier function as follows:

$$d_l = (1 - t)^2 p_0 + 2(1 - t)t p_1 + t^2 p_2 \quad (5)$$

such that

$$t = (m_l - m_n)/(m_c - m_n) \quad (6)$$

In equation 5, p_0 is the bottom neck depth, p_2 is the chest depth, and p_1=p_2-1. In equation 6, m_l is the midpoint of the upper torso line in process, m_n is the midpoint of the neck bottom line, and m_c is the midpoint of the chest line. To estimate the lower torso depth, the system first calculates the hip depth d_h, which is the average depth of the thigh lines. In addition, it retrieves m_1, the midpoint of the thigh lines. After that, for each lower torso line, the system calculates the depth d_l by linearly interpolating between d_c and d_h as follows:

$$d_l = (r * (d_h - d_c)) + d_c \quad (7)$$

such that

$$r = (m_l - m_c)/(m_1 - m_c) \quad (8)$$

In equation 8, m_l is the midpoint of the torso line in process, and m_c is the midpoint of the chest line.

3D Mesh Construction & Smoothing

To create the character's mesh, the system converts 2D and depth data into a 3D structure that defines the shape of the character. To do that, the system uses the relative position and orientation of each body section line and converts it into a 3D profile curve, reflecting its calculated depth. Using profile curves, the system constructs polygon surfaces on which edge loops are formed to reflect the position and orientation of those curves. Fig. 8 shows front and side views of the polygon surfaces.

To complete the head mesh, the system attaches a new surface, subdivided for suitable edge flow, on each end of the head. The system then attaches the head bottom to the neck by bridging between the neck's top edge loop and the head bottom's back edges, deleting unnecessary surface faces. To smooth the head top and joined head-neck area, the system implements a series of Bezier curves.

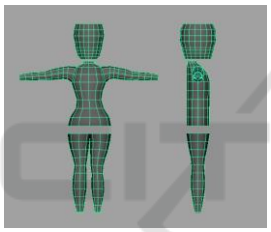

Figure 8: Polygon surfaces constructed using 2D coordinates and orientation, as well as depth data.

To connect the torso to the legs, the system partially bridges between the thigh lines and the bottom edge of the torso, leaving the middle section unconnected. When constructing the middle surface, the system enforces a minimal distance of 0.1 centimeters between the inner leg vertices to prevent overlapping geometry. After that, the inner legs and torso are joined by a single surface, subdivided for suitable edge flow. Finally, the system implements a series of quadratic Bezier curves (equation 5) to smooth the geometry of the hips. In addition, the system curves two edges, using their vertices, around the hips to follow the joint and muscle structure.

Connecting shoulders to the torso, the last process the system automatically applies, is different as the top shoulder line is already aligned with the upper torso lines from both sides (Fig. 8). Thus, the system averagely merges aligned shoulder and upper torso vertices, eliminating torso side faces bounded by those vertices.

The result is a connected mesh that is smoothed for better appearance and enhanced geometry. Our algorithm strives to follow the standard and recommended guidelines in modeling human characters, producing suitable geometry and topology. Fig. 9 shows a complete 3D mesh of the character. In addition, Fig. 10 shows a sample of two sketches along with the system-generated mesh. Note the cartoon-like proportions (matching the sketches) as well as the proper edge flow around the shoulders, hips and neck.

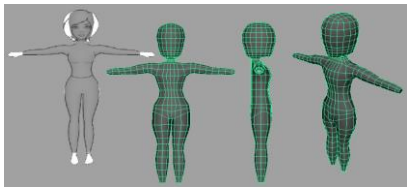

Figure 9: 3D character mesh. From left to right, the first two are in front view. The third and fourth are in side and perspective views, respectively.

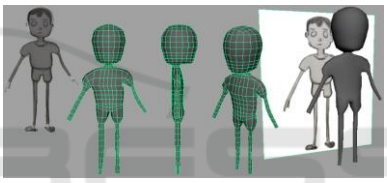

Figure 10: Sample of a system-generated mesh.

4 EXPERIMENTS AND RESULTS

4.1 User Study

We conducted a user study to test our system with a variety of sketched characters and collect feedback from 3D artists. We asked participants to bring sketches that met the requirements listed earlier in this paper. After a short demonstration, we proceeded with the study as follows:

Using their character sketches, participants used the UI to input the 10 required points on the character, and had the system automatically produce the 3D character mesh. The system logged the time durations of user interaction, 2D sketch processing and 3D mesh construction, and the users completed a post-study questionnaire.

We had a total of 15 participants, all below the age of 40. The system succeeded in producing a mesh with 14 out of the 15 participant sketches. Users rated their level of experience in 3D modeling generally, and in character modeling particularly. In 3D modeling, 7% said they never modeled before,

29% rated themselves as beginners, 21% as intermediate, 36% as advanced, and 7% as expert. On the other hand, in character modeling, 14% said they never modeled characters before, 43% were beginners, 29% intermediate, and 14% advanced. Advanced artists estimated that a similar mesh would take them between 30 minutes and an hour to create.

User Evaluation

Regarding the sketch specifications required by the system, 29% of the users said the specifications were not enough, suggesting that an added side-view sketch would create better depth results. 57% said the amount was enough, expressing that the requirements were "reasonable" and "straightforward". 14% said "a lot" because they limit the flexibility to accept additional details like "weapons or other pieces of armor". On the other hand, 7% of users said that the 10-point input was too little, 86% said it was enough, and 7% said it was too much, as input points could be mirrored when using symmetric characters. When asked to rate the learning curve and ease of interaction with our UI, 100% of users agreed or strongly agreed that the system was easy to learn and to use. All users also agreed or strongly agreed that the system generated the mesh quickly.

86% of the users said that they would use the generated mesh to build their 3D character. Overall, comments about the system results were positive. Users' description of the system included "very useful", "quick", "efficient", and "accurate size estimation". Their suggestions included "more flexibility" and "more depth control".

4.2 Discussion

The results of the user study show promising feedback regarding our system. All users agreed that the UI is easy to understand and interact with. Users stated that it was "straightforward" and "very easy to manipulate", although one user suggested changing the "wizard style interface", and another suggested that it would be easier to "pre-generate" adjustable points on the sketch. Users also agreed that the system generated the mesh quickly. Most users also highly rated the geometry and topology of the mesh, and expressed their willingness to use the mesh to construct the character model. On the other hand, a few users criticized the limitations of a system that does not recognize additional objects like hats or bags, and is less accurate with non-fitted garment.

Our system generated the mesh in an average of 11.5 seconds, including 2D and 3D processing. The approximate average time of user interaction was 2 minutes and 13 seconds. We did not find a significant correlation between manual construction time and user experience, and it was interesting to see that the approximated manual time to block out the basic geometry of the character does not decrease as modelers become more experienced.

Finally, part of our research questions was to determine if users value the speed of an automatic SBM system. Users' preference in using an automatic system versus manual modeling did not greatly change when the system hypothetically lost its speed advantage. Novice modelers (beginners) leaned towards automatic modeling regardless of the time it takes, which supports Alhashim's (2015) statement about how novice modelers find it less tedious to work with existing meshes rather than starting from scratch. On the other hand, intermediate and advanced modelers were more likely to base their decision to use the automatic system on its capabilities and the specific details it produces in a mesh.

5 CONCLUSIONS

In this paper, we proposed a system that automatically creates the basic geometry of a 3D human character using a single-view sketch and minimal user input. Our goal was to create an algorithm that conforms to the current modeling techniques as well as the geometry and topology that professional modelers recommend. User evaluation of the mesh indicates that our algorithm produces good and usable meshes. The user study conducted to evaluate this system resulted in positive feedback from both beginner and intermediate users who found it helpful and quick to start up the process.

5.1 Limitations & Future Work

Future work could improve the method that locates the clavicle line, as we encountered issues when the character's arms were not fully extended in a T-pose. In addition, to reduce the complexity of adjusting head measurements, our system produces simple geometry that is easy to manually fix. However, for more accurate results, the system could use internal strokes of the sketched character to trace the hairline and estimate the hair volume.

In conclusion, despite the limitations of our system, results and feedback indicate that our system produces suitable results for non-advanced modelers to speed up the modeling process, using an intuitive

and easy to use UI. 3D modeling students showed interest in using the system, and modeling experts expressed the usefulness of the system to beginners. This proves that our system contributes is a step towards building more familiar and supported SBM system.

ACKNOWLEDGEMENTS

This project was funded by The Interactive and Multi-Modal Experience Research Syndicate (IMMERSe).

REFERENCES

Alexe, A., Gaildrat, V. and Barthe, L., 2004, November. Interactive modelling from sketches using spherical implicit functions. *In Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa (pp. 25-34).* ACM.

Alhashim, I., 2015. Modeling and Correspondence of Topologically Complex 3D Shapes.

Andre, A. and Saito, S., 2011, August. Single-view sketch based modeling. *In Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling (pp. 133-140).* ACM.

Bessmeltsev, M., Chang, W., Vining, N., Sheffer, A. and Singh, K., 2015. Modeling character canvases from cartoon drawings. *ACM Transactions on Graphics (TOG), 34(5), p.162.*

Churchill, E., McConville, J.T., Laubach, L.L., Erskine, P., Downing, K. and Churchill, T., 1978. Anthropometric source book. *Volume II: handbook of anthropometric data.*

Buchanan, P., Mukundan, R. and Doggett, M., 2013, July. Automatic single-view character model reconstruction. *In Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling (pp. 5-14).* ACM.

Celebi, M.E., 2011. Improving the performance of k-means for color quantization. *Image and Vision Computing, 29(4), pp.260-271.*

Cook, M.T. and Agah, A., 2009. A survey of sketch-based 3-D modeling techniques. *Interacting with computers, 21(3), pp.201-211.*

Dargie, J., 2007. Modeling techniques: movies vs. games. ACM SIGGRAPH Computer Graphics, 41(2), p.2.

Entem, E., Barthe, L., Cani, M.P., Cordier, F. and Van de Panne, M., 2015. Modeling 3D animals from a side-view sketch. *Computers & Graphics, 46, pp.221-230.*

Gonen, O. and Akleman, E., 2012. Sketch based 3D modeling with curvature classification. *Computers & Graphics, 36(5), pp.521-525.*

Igarashi, T., Matsuoka, S. and Tanaka, H., 2007, August. Teddy: a sketching interface for 3D freeform design. *In Acm siggraph 2007 courses (p. 21).* ACM.

Johnston, A., Carneiro, G., Ding, R. and Velho, L., 2015, November. *3-D Modeling from Concept Sketches of Human Characters with Minimal User Interaction. In Digital Image Computing: Techniques and Applications (DICTA), (pp. 1-8).* IEEE.

Karpenko, O.A. and Hughes, J.F., 2006, July. SmoothSketch: 3D free-form shapes from complex sketches. *In ACM Transactions on Graphics (TOG) (Vol. 25, No. 3, pp. 589-598).* ACM.

Kazmi, I.K., You, L. and Zhang, J.J., 2014, August. A survey of sketch based modeling systems. In Computer Graphics, Imaging and Visualization (CGIV), *2014 11th International Conference on (pp. 27-36).* IEEE.

Lin, Y.L. and Wang, M.J.J., 2011. Automated body feature extraction from 2D images. *Expert Systems with Applications, 38(3), pp.2585-2591.*

Mao, C., Qin, S.F. and Wright, D., 2009. A sketch-based approach to human body modelling. *Computers & Graphics, 33(4), pp.521-541.*

Mao, C., Qin, S.F. and Wright, D.K., 2006. *Applying scenarios in user-centred design to develop a sketching interface for human modelling and animation.* Eurographics.

Nealen, A., Igarashi, T., Sorkine, O. and Alexa, M., 2007. FiberMesh: designing freeform surfaces with 3D curves. *ACM transactions on graphics (TOG), 26(3), p.41.*

Olsen, L., Samavati, F.F., Sousa, M.C. and Jorge, J.A., 2009. Sketch-based modeling: A survey. *Computers & Graphics, 33(1), pp.85-103.*

Patnode, J., 2012. *Character modeling with Maya and ZBrush: professional polygonal modeling techniques*. CRC Press.

Rivers, A., Durand, F. and Igarashi, T., 2010. *3D modeling with silhouettes (Vol. 29, No. 4, p. 109).* ACM.

Selinger, P., 2003. *Potrace: a polygon-based tracing algorithm.* Potrace (online), http://potrace. sourceforge. net/potrace. pdf (2009-07-01).

Viana, R.D., Nascimento, E.R. and Ferreira, R.A., 2014. *On the Development of a Fully Automatic Methodology to Create Smooth Mesh for 3D Models from Concept Arts.*

Villar, O., 2014. *Learning Blender: a hands-on guide to creating 3D animated characters.* Addison-Wesley Professional.

Zhou, X., Chen, J., Chen, G., Zhao, Z. and Zhao, Y., 2016. Anthropometric body modeling based on orthogonal-view images. *International Journal of Industrial Ergonomics, 53, pp.27-36.*