Received September 29, 2014; Accepted January 4, 2015

International Journal of Asia Digital Art&Design

2.5D Modeling from Illustrations of Different Views

Kitamura, Maki
Kyushu University
maki.terai@gmail.com

Kanamori, Yoshihiro
University of Tsukuba
kanamori@cs.tsukuba.ac.jp

Tsuruno, Reiji
Kyushu University
tsuruno@design.kyushu-u.ac.jp

Abstract
When artists design characters, they draw illustrations viewed from the front, side or slant views. In this paper, we create a 2.5D model using such illustrations. A 2.5D model is a model that is created by arranging deformable billboards along the depth direction, and it can express appearance of the character between two viewpoints with considering depth information. Our method uses two cartoon-like illustrations and the corresponding eye directions as inputs. These illustrations are composed of contours and closed regions painted with uniform colors. Given closed regions in illustrations, our method finds corresponding closed regions between two illustrations based on an improved similarity function, which yields better matching than a previous method. After obtaining correspondences of closed regions, our method creates a 2.5D model by transforming each pair of matched regions as a 3D billboard whose position can be estimated based on the correspondence. These processes are semi-automatic. The user then assigns feature points manually along the corresponding contours of matched regions so that the contours can be interpolated naturally. Finally, by completing regions that appear or vanish with user strokes, our method can interpolate illustrations with occluded regions. We demonstrate that our method can create 2.5D models with various illustrations.

Keywords: 2.5D modeling, billboards, shape matching

1 Introduction

When artists design a character, they draw illustrations viewed from the front, side or slant views. In a 3D production, designers create a 3D model with these illustrations drawn from different views. However, 3D modeling takes much time, and expressing a nuance of 2D illustrations is difficult with a 3D model. In order to solve these problems, 2.5D modeling can be used [1], [2]. A 2.5D model is a model that can express appearance among multiple views using many layers segmented in many parts such as eyes and a mouth, and have depth information, while keeping the nuance of 2D illustrations in a 3D space. Using the system of Rivers et al. [2], the user can create a 2.5D model from scratch by drawing each part and warping the part for each view manually, which is time-consuming.

In this paper, we propose a novel 2.5D modeling workflow that reduces the labor for creating parts, by diverting the illustrations drawn in character design. The most related methods to our work are morphing techniques [3], [4], [5] that can interpolate multiple images. However, these methods cannot interpolate occluded parts. On the other hand, our method can interpolate such parts by completing occluded regions with user strokes. We confirmed that our method can create 2.5D models with various illustrations.

2 Related Work

In the field of 3D computer graphics, much research that expresses 3D models in a hand-drawn illustration style has developed in non-photorealistic rendering (NPR) [6], [7]. While generating 2D images in hand-drawn styles from 3D models has been researched widely, the opposite is difficult. In other words, creating a 3D model from hand-drawn illustrations of different views is difficult because illustrations drawn from different views may be inconsistent. View-dependent geometry [8] tackled this problem. This method enables view-dependent shape appearance that looks inconsistent if the shape were a still object. This is accomplished by warping a 3D model in each view and interpolating them linearly, which requires additional 3D modeling in each view. Our method enables such view-dependent expression using a 2.5D model.

There is much research for 2D character animations. Igarashi et al. [9] proposed a method that enables a character animation by creating a 2D mesh from a 2D illustration and deforming the mesh while keeping features of the shape. Applying motion capture data to a single character illustration based on a skeleton structure [10], [11] is also studied. To make character animations by using multiple images, morphing methods [3], [4], [5] have been proposed. Baxter et al. [12] tried to animate hand-drawn illustrations with morphing. Although these methods can handle appearance of characters in a fixed view, they cannot handle occlusions caused by changes of the viewpoint. This is because these methods cannot find correspondence required for interpolation at missing parts due to occlusion.

Whited et al. [13] proposed a method for inbetweening of hand-drawn animations. Their method can interpolate partially occluded strokes by letting the user draw missing parts and to assign correspondence manually. However, their method cannot handle large changes of views. Furusawa et al. [14] focused on interpolation of two illustrations of a character's face by matching feature points based on pre-defined rules, and enabled expression of occluded regions. However, their method cannot handle a situation where regions are partially occluded because their method does not consider depth information. Our method can handle such the situation.

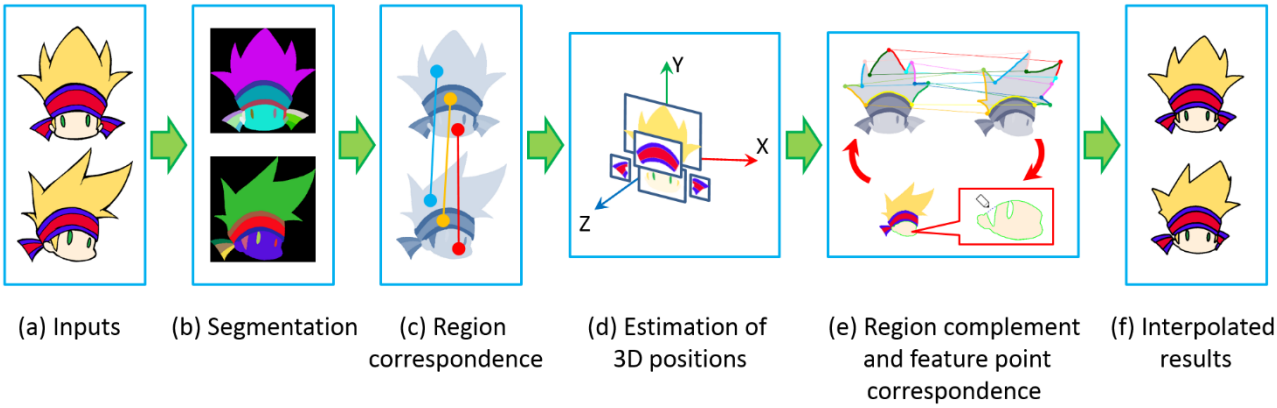

Figure 1 System overview.

To create 2.5D models, Di Fiore et al. [1] proposed a system where the user specifies depth information manually. On the other hand, our method estimates depth information based on the viewing direction of each view automatically. Rivers et al. [2] proposed an interface to create 2.5D models. Although their method can estimate depth information of each part automatically, users need to create each part from scratch. Our method can divert illustrations drawn at the time of character design, saving the time to create parts. Yeh et al. [15] proposed a method that enables novel operations including rolling, twisting and folding by mixing elements from both the front and back sides of images. Their method cannot handle expressions of appearances of different views. On the other hand, our method can handle such expressions.

3 Algorithm

3.1 System Overview

Our system uses two character illustrations and corresponding view directions. We set a 3D right-handed orthogonal coordinate system as follows (Figure 2). We set the center of a character as the origin, character's gazing direction as the positive direction of *z* axis, its left-hand side as the positive direction of *x* axis, and its upper side as the positive direction of *y* axis, respectively. Let $\mathbf{V}_{view}$ be observer's view direction. Let θ be the angle between $\mathbf{V}_{view}$ and *z* axis in *xz* plane, and let φ be the angle between $\mathbf{V}_{view}$ and *z* axis in *yz* plane. A front view then corresponds to $(\theta, \varphi) = (0,0)$.

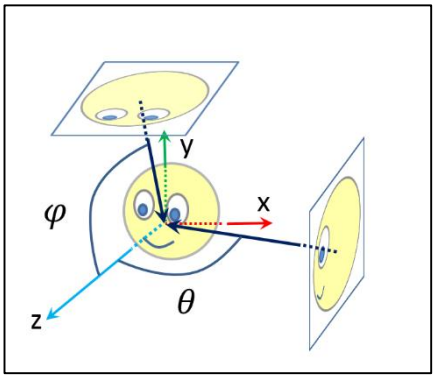

Figure 2 Right-hand coordinate system and view directions.

Figure 1 shows the overview of our system. The input illustrations are assumed to have black contours and closed regions painted with uniform colors (Figure 1(a)). Our system segments each illustration into closed regions (Figure 1(b)), and then finds region-wise correspondence between the two illustrations (Figure 1(c)). These closed regions are arranged in 3D space as billboards, and the 3D positions are estimated from the centroids of matched regions and the view directions (Figure 1(d)). The billboards are displayed using parallel projection. The system interpolates billboard shapes linearly between the two illustrations. Prior to interpolation, users complete the shape of closed regions and assign feature points manually along the corresponding contours of matched regions so that the contours can be interpolated naturally (Figure 1(e)). In this way, the system can create a 2.5D model and handle appearance changes with consideration of depth information.

3.2 Region Segmentation and Complement

At first, our system segments input illustrations into closed regions by using the flood fill algorithm (Figure 1(b)). Note that contours must be also assigned depth values; otherwise gaps appear between closed regions during interpolation. The system integrates contour lines into closed regions using Sýkora et al.'s method [16]. Specifically, we integrate each pixel of contour lines into the nearest closed region (Figure 3). Moreover, if a small closed region lies inside a large closed region, the small region causes a hole in the large region, and the hole will be

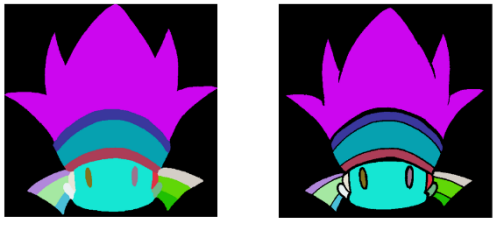

Figure 3 Closed regions (a) with and (b) without integration of contour lines.

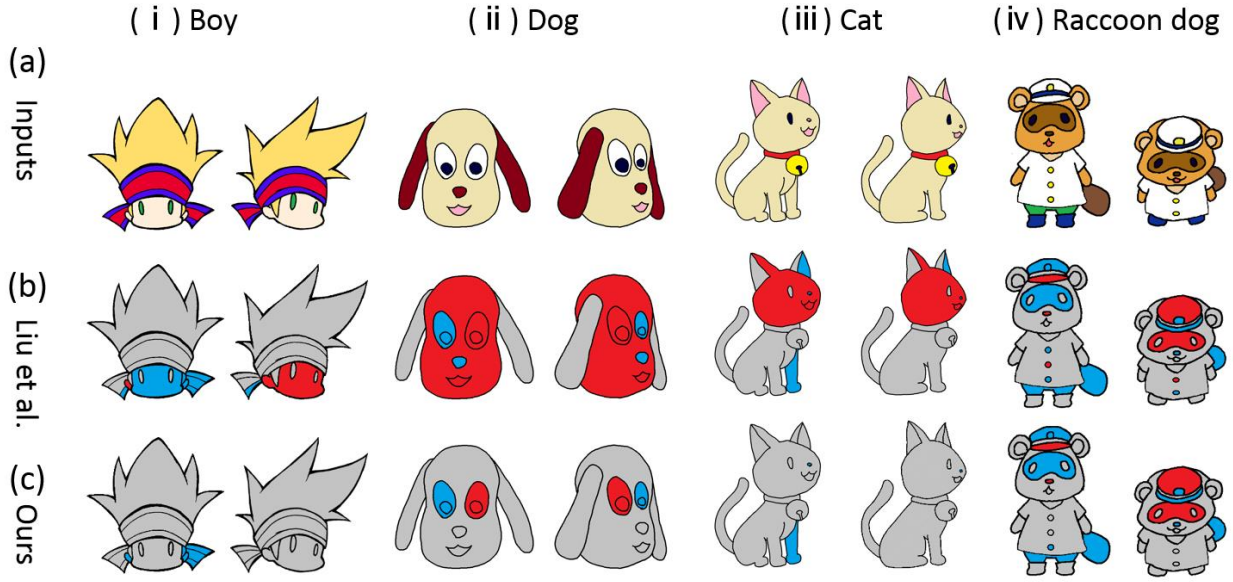

Figure 4 Comparison of results of region correspondence. Gray regions indicate correct correspondences, red regions wrong correspondences, and blue regions no correspondences.

interpolated unless it is completed. Our system therefore com-

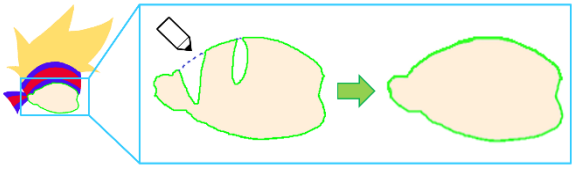

Figure 5 Region complement. Users can complete a region by drawing strokes.

pletes such holes automatically. What is more, a 2.5D model may exhibit gaps between adjacent parts even if the parts are adjacent to each other in the illustration because of depth difference of billboards and shape deformation caused by projection. To reduce such gaps, users can modify the shape of occluded regions by drawing strokes (Figure 5).

3.3 Region Correspondence with Similarity

After segmenting regions, our system finds corresponding regions between two illustrations based on a similarity function (Figure 1(c)). Liu et al. [17] also proposed a similarity function similar to ours. In their method, a similarity of two regions, *a* and *b*, is calculated in terms of their differences in color, shape and size. However, in case that one region lies inside another region, the similarity of two regions becomes maximum erroneously. This can become a problem in our system. For example, if an eye region in one illustration falls into a face region in the other illustration, the similarity of the eye region and the face region becomes maximum and the two regions are matched in error. Therefore we propose an improved similarity function of two regions *a* and *b* as follows.

$$s_{a,b} = \mathcal{J}_{a,b} e^{-w_x|x_a - x_b|} e^{-w_y|y_a - y_b|} e^{-\frac{|S_a - S_b|}{(S_a + S_b)/2}} \quad (1)$$

where $\mathcal{J}_{a,b}$ is the same as Liu's method. This distinguishes whether a color difference of *a* and *b* is within a certain range.

$$\mathcal{J}_{a,b} = H[T_c - C_{a,b}] \quad (2)$$

where $C_{a,b} = \|\mathbf{q}_a - \mathbf{q}_b\|$ is the Euclidean norm of $\mathbf{q}_a$ and $\mathbf{q}_b$. $\mathbf{q}_a$ and $\mathbf{q}_b$ are the colors of closed regions *a* and *b* in the RGB color space. T_c is an user-defined threshold and set as 0.3 in our system. $H[n]$ is the Heaviside step function, namely, it becomes 0 if n is negative and 1 otherwise. $e^{-w_x|x_a - x_b|} e^{-w_y|y_a - y_b|}$ is a similarity of positions, and calculated with consideration of view directions and positions of closed regions. Let (x_a, y_a) and (x_b, y_b) be the centroids of bounding boxes of *a* and *b*. If the angle θ does not change (i.e., the view direction changes only vertically) and $|x_a - x_b|$ is small, the similarity should be large. Similarly, if the angle φ does not change and $|y_a - y_b|$ is small, the similarity should be large. We set $w_x = 1$ if $\theta = 0$ otherwise $w_x = 0.5$. Similarly, we set $w_y = 1$ if $\theta = 0$ otherwise $w_y = 0.5$. $e^{-\frac{|S_a - S_b|}{(S_a + S_b)/2}}$ is a similarity about sizes of closed regions. Let S_a and S_b be areas of closed regions *a* and *b*. If values of S_a and S_b are close to each other, the similarity becomes high. To normalize the similarity, we divide it by the average of their areas.

Our similarity function is used as follows. For each closed region in one illustration, the system finds the closed region that has the highest similarity value in the other illustration, and makes a correspondence between the two regions. Figure 4 and Table 1 show the results of matching experiments. Figure 4

shows (a) input images, (b) results of Liu et al. and (c) our result. The gray regions indicate correct correspondences, red regions wrong correspondences, and blue regions no correspondences. In the boy example, the results of Liu et al. do not have correct correspondences for the ear and the face. This is a typical error of their method as mentioned above. In our results, although there are no correspondences for both the ears and the part of the right knot of the headband, this is correct because actually they do not have corresponding regions. In the dog example, results of Liu et al. are wrong in the face and the mouth, and each mouth is mismatched with each face as the typical error. For the both eyes of the dog, both Liu et al.'s and our results are wrong, and each left eye is mismatched with each right eye. In the cat example, results of Liu et al. are wrong in the face and the mouth. Both our and Liu et al.' results have no matched region with the right forefoot, but this is correct. In a button region of the raccoon dog example, Liu et al.'s result is wrong, but our result is correct. For another wrong results in the raccoon dog, Liu et al.'s results are the same as our results. If correspondences are wrong, users can modify the correspondences interactively. Moreover, if there is no correspondence, users can create a corresponding region by drawing strokes. Table 1 summarizes the statistics of both Liu et al.'s and our matching results.

Table 1 The numbers of closed regions that are matched correctly with the regions that do not disappear and have corresponding regions. "# of regions" is the numbers of closed regions that have corresponding regions. The numbers in parentheses indicate the percentages of correct results.

Illustrations	# of regions	Liu et al.	Our method
Boy	11	9 (82%)	11 (100%)
Dog	9	2 (22%)	5 (56%)
Cat	12	8 (67%)	11 (92%)
Raccoon dog	23	15 (65%)	17 (74%)

3.4 Estimation of 3D Billboard Positions

After obtaining correspondences of closed regions, our system can estimate a 3D billboard's position $\mathbf{p}$ (Figure 1(d)). To estimate $\mathbf{p}$, we use Rivers' method [2]. We confirmed that they use Algorithm 1 via a personal communication.

Algorithm 1 Calculate a 3D billboard position.

$\mathbf{p}_{sum} \leftarrow (0,0,0), \mathbf{p}_{current} \leftarrow (0,0,0)$
$N \leftarrow$ many times (e.g., 10,000)
for $i = 0$ to N do
  $N_v \leftarrow$ the number of views
  for $j = 0$ to N_v do
    $c_j \leftarrow$ the center of bounding box of region R_j
    $l_j \leftarrow$ the 3D line that passes through c_j and goes in the view direction of the view
    $\mathbf{p}_j \leftarrow$ the 3D position that projected $\mathbf{p}_{current}$ onto l_j
    $\mathbf{p}_{sum} \leftarrow \mathbf{p}_{sum} + \mathbf{p}_{current}$
  end for
  $\mathbf{p}_{current} \leftarrow \mathbf{p}_{sum}/N_v$
end for
$\mathbf{p} \leftarrow \mathbf{p}_{current}$

When no interpolation is performed, the projected position of each billboard should coincide with its original positions in input illustrations. However, because Algorithm 1 never guarantees the coincidence, projected positions are slightly shifted. To eliminate such shifts, our system calculates the initial and terminal 3D positions of each billboard as follows. We first calculate a 3D position using Algorithm 1, and then project it onto each half line that passes through the centroid of a corresponding region along a view direction of an input illustration. Our system then interpolates the 3D position of a billboards linearly between the initial and terminal 3D positions. Moreover, our system allows the user to further arrange the depth of each billboard along view directions interactively.

3.5 Correspondence of Feature Points

After the estimation of a 3D position, the shape of each billboard is interpolated between its original shapes in input illustrations. The contours of two matching regions are cut into curve segments at feature points, and are linearly interpolated in the curve segment basis. To obtain such feature points, we implemented an automatic method by Baxter et al. [18], but we found that it is error-prone and actually increases the burden of finding mismatches. Consequently, we decide to let the user assign feature points manually.

4 Results

We implemented our prototype system using C++ and Qt library, and ran it on a PC with Intel Core i7-2760QM 2.40GHz CPU. Figure 6 shows our results. The illustrations in red frames are input images and the illustrations in blue frames are synthesized frames by our system. For each example, we used illustrations of a front view and another view. In the examples of the boy, the dog and the cat, we used illustrations where only θ changes by 45 degrees. In the raccoon dog example, we used illustrations where only φ changes by 45 degrees. The time required for creating each 2.5D model was in the range between about 10 minutes and about 50 minutes, where most of time was spent for manual complement of regions and manual modification of depth values. Our system could successfully handle the following examples where occlusion occur; the occluded knot of the headband in the boy example, the occluded left ear in the dog, the occluded left forefoot in the cat, the occluded lowest yellow button and the pants in the raccoon dog.

Our current implementation has the following limitations. Our prototype interpolates only region contours but eliminates decorative lines such as those in boy's hair or around the mouths of the cat, dog and raccoon dog. Our prototype also cannot express the variations of line thickness. These are not the limitations of our method but those of our implementation, which we would like to eliminate in future work.

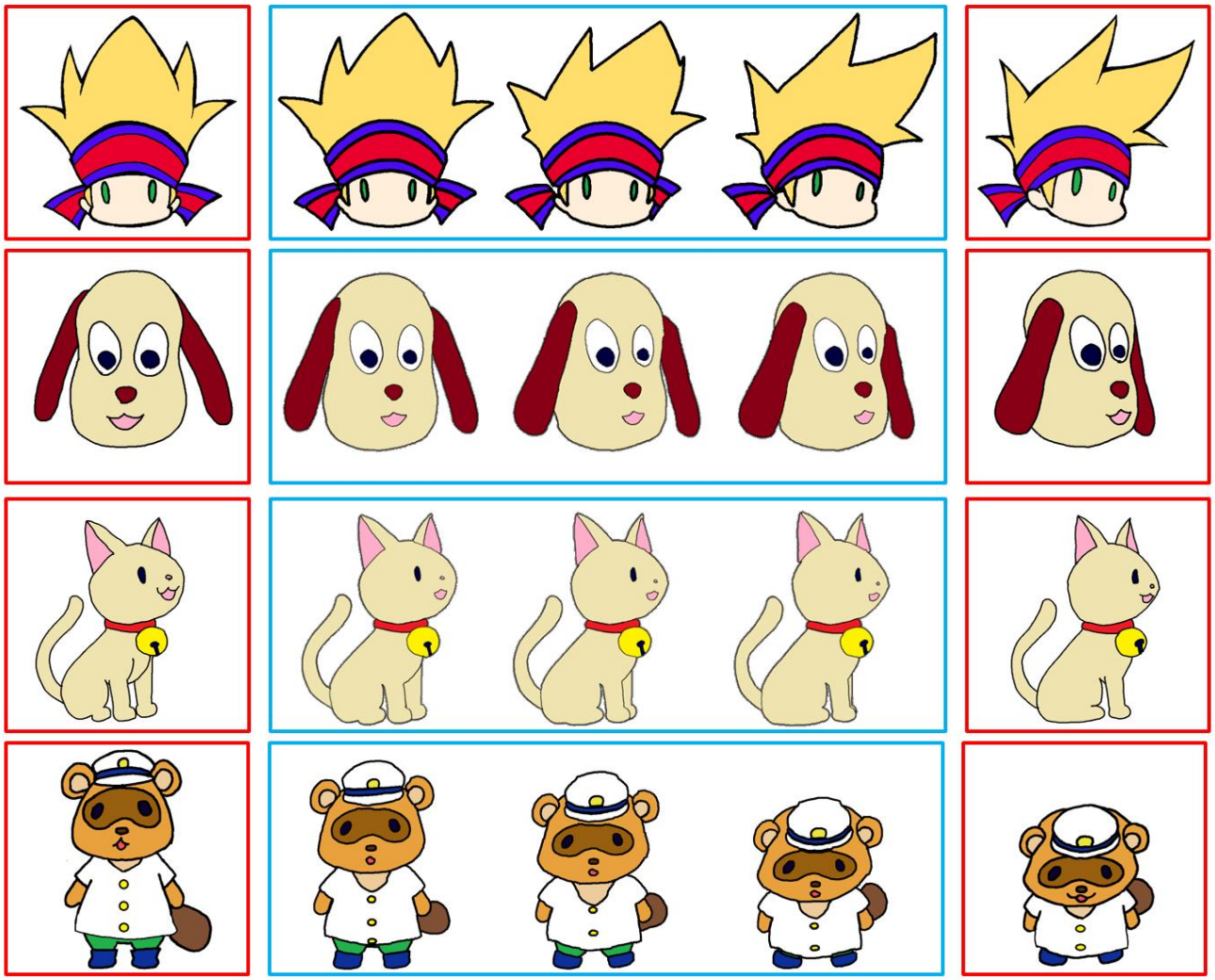

Figure 6 Interpolation results synthesized by our system. The illustrations in red frames are input images and the illustrations in blue frames are synthesized frames by our system.

5 Conclusion

In this paper, we have proposed a method to create 2.5D models from two cartoon-like illustrations of different views. First, our system segments regions in input illustrations into closed regions, and calculates matchings of regions between two illustrations using an improved similarity function. Next, the system creates a 2.5D model by arranging closed regions in a 3D space as billboards. Our system reduces the burden of making a 2.5D model by utilizing illustrations as inputs. As a limitation, our method does not consider the depth order of regions in input illustrations when estimating 3D positions of billboards, which causes some billboards to be occluded by others incorrectly. For example, an eye region that should be in front of a face region may be occluded by the face region in 3D space. To reduce such errors, our system needs to take account of relative distances of other billboards when estimating 3D positions. In future work, we would like to propose a method that can estimate 3D positions with consideration of such relative distances, and users can edit such relative distances easily. Moreover, we would like to propose a system to create arbitrary poses of characters by extending our system.

Acknowledgement

This work was supported by the Japan Society for the Promotion of Science (JSPS).

References

[1] Di Fiore, F., Schaeken, P., Elens, K. and Van Reeth, F, *Automatic in-betweening in computer assisted animation by exploiting 2.5D modelling techniques*, Computer Animation, pp. 192-200, 2001.

[2] Rivers, A., Igarashi, T. and Durand, F., *2.5D Cartoon Models*, ACM Trans. Graph., Vol. 29, No. 4, pp. 59:1-59:7, 2010.

[3] Baxter, W., Barla, P. and Anjyo, K.-i., *Rigid Shape Interpolation Using Normal Equations*, Proceedings of the 6th International Symposium on Non-photorealistic Animation and Rendering, pp. 59-64, 2008.

[4] Sýkora, D., Dingliana, J. and Collins, S., *As-rigid-as-possible Image Registration for Hand-drawn Cartoon Animations*, Proceedings of International Symposium on Non-photorealistic Animation and Rendering, pp. 25-33, 2009.

[5] Wolberg, G., *Image morphing: a survey*, The Visual Computer, Vol. 14, No. 8-9, pp. 360-372, 1998.

[6] Decaudin, P., *Cartoon Looking Rendering of 3D Scenes*, Research Report 2919, INRIA, 1996.

[7] Rusinkiewicz, S., DeCarlo, D. and Finkelstein, A., *Line Drawings from 3D Models*, ACM SIGGRAPH 2005, Courses, 2005.

[8] Rademacher, P., *View-dependent Geometry*, Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99, pp. 439-446, 1999.

[9] Igarashi, T., Moscovich, T. and Hughes, J. F., *As-rigid-as-possible Shape Manipulation*, ACM Trans. Graph., Vol. 24, No. 3, pp. 1134-1141, 2005.

[10] Hornung, A., Dekkers, E. and Kobbelt, L., *Character Animation from 2D Pictures and 3D Motion Data*, ACM Trans. Graph., Vol. 26, No. 1, 2007.

[11] Pan, J. and Zhang, J., *Sketch-Based Skeleton-Driven 2D Animation and Motion Capture*, Transactions on Edutainment VI, Lecture Notes in Computer Science, Vol. 6758, Springer Berlin Heidelberg, pp. 164-181, 2011.

[12] Baxter, W., Barla, P. and Anjyo, K., *N-way Morphing for 2D Animation*, Computer Animation and Virtual Worlds, Vol. 20, No. 2-3, pp. 79-87, 2009.

[13] Whited, B., Noris, G., Simmons, M., Sumner, R., Gross, M. and Rossignac, J., *BetweenIT: An Interactive Tool for Tight Inbetweening*, Comput. Graphics Forum, Vol. 29, No. 2, pp. 605-614, 2010.

[14] Furusawa, C., Fukusato, T., Okada, N., Hirai, T. and Morishima, S., *Quasi 3D Rotation for Hand-drawn Characters*, ACM SIGGRAPH 2014 Posters, SIGGRAPH '14, pp. 12:1-12:1, 2014.

[15] Yeh, C.-K., Song, P., Lin, P.-Y., Fu, C.-W., Lin, C.-H. and Lee, T.-Y., *Double-Sided 2.5D Graphics*, IEEE Transactions on Visualization and Computer Graphics, Vol. 19, No. 2, pp. 225-235, 2013.

[16] Sýkora, D., Buriánek, J. and Žára, J., *Sketching Cartoons by Example*, Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling, pp. 27-34, 2005.

[17] Liu, X., Mao, X., Yang, X., Zhang, L. and Wong, T.-T., *Stereoscopizing Cel Animations*, ACM Transactions on Graphics (SIGGRAPH Asia 2013 issue), Vol. 32, No. 6, pp. 223:1-223:10, 2013.

[18] Baxter, W., Barla, P. and Anjyo, K.-i., *Compatible Embedding for 2D Shape Animation*, IEEE Transactions on Visualization and Computer Graphics, Vol. 15, No. 5, pp. 867-879, 2009.