Automatic 2.5D Cartoon Modelling

Fengqi An
School of Computer Science
and Engineering
University of New South Wales
NSW 2052, Australia
Email: fan@cse.unsw.edu.au

Xiongcai Cai
School of Computer Science
and Engineering
University of New South Wales
NSW 2052, Australia
Email: xcai@cse.unsw.edu.au

Arcot Sowmya
School of Computer Science
and Engineering
University of New South Wales
NSW 2052, Australia
Email: sowmya@cse.unsw.edu.au

***Abstract*—2.5D cartoon modelling is a recent technique used for modelling 2D cartoons in 3D, to enable 2D cartoons to be rotated and viewed in 3D. Automatic modelling is essential to efficiently create 2.5D cartoon models. However, existing approaches to 2.5D modelling are based on manual 2D drawings by artists, which is inefficient and labour intensive. In this paper, we present a framework for automatic 2.5D cartoon modelling from 3D models, which provides an automatic approach to construct 2.5D cartoon models. Our experiments show that the proposed method is efficient and automatic.**

I. INTRODUCTION

The term "2.5D" is often used to describe the techniques used in video games and animations to simulate 3D scenes using only 2D elements. These include games based on isometric tiles and parallax scrolling where the viewing angles are often fixed. Some other 2.5D methods [4], [2] provide for viewpoint changes, but these methods are relatively limited as they rely heavily on information manually provided by the artists, and thus are inefficient and labour intensive. Recently, Rivers et al. [10] proposed a novel solution for full 3D rotation of 2D drawings, called "2.5D Cartoon Models", which combines interpolation and rendering methods. More importantly, it retains 2D stylistic strokes while rotating in 3D, thereby enabling presentation of elements that are impossible to define using current 3D techniques.

Currently, the building process for such models requires the user to have sufficient drawing abilities and could take a great amount of time and effort as the user actually needs to manually draw a number of 2D cartoons from different views. Defining a curve in 2.5D normally requires much more work than defining it in 2D vector graphics, because a regular 2.5D cartoon model is often defined in 3 or more different 2D planes. The manual process makes the creation of 2.5D cartoon models not only inefficient but also labour intensive. Therefore, an automatic system that allows users to quickly build up a 2.5D model is essential for improving the usability of such models.

In this paper, we propose a framework to automatically construct 2.5D cartoon models from 3D models. Creating a new 2.5D cartoon model from an existing 3D model using our system requires very few human operations. These operations normally take less than 3 minutes. While creating a similar model manually from scratch requires the user to put in a lot of effort and is very time consuming. On the other hand, our system does not fully take over the role of the artist. Instead, it is a tool to help artists sketch up a 2.5D model, which helps to quickly build models by modifing the sketches.

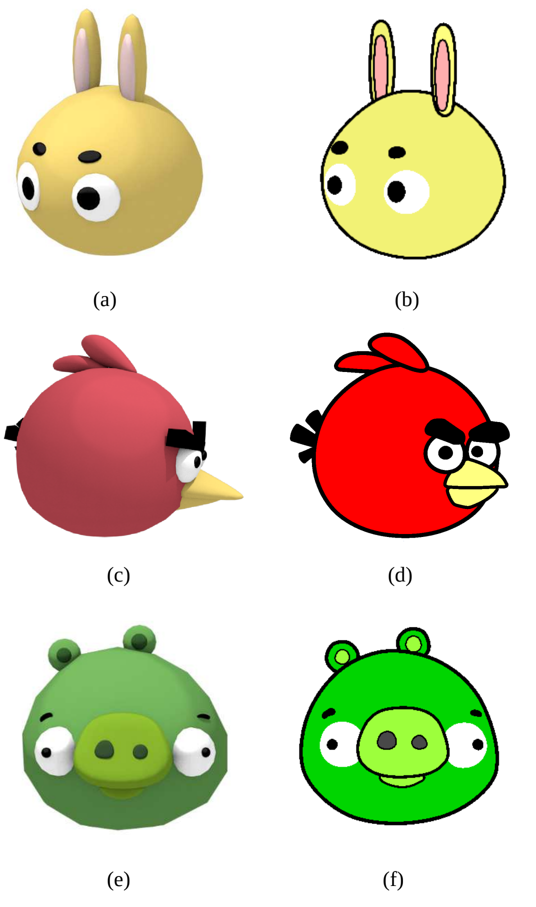


Fig. 1. (a): A 3D model of little rabbit 'Ruby'. (b): 2.5D cartoon Ruby. (c): A 3D Angry Bird. (d): 2.5D cartoon bird. (e): A 3D pig. (f): 2.5D pig.

The paper is organized as follows. Related work is surveyed in Section II. 2.5D cartoon models are described in Section III. The proposed automatic 2.5D modelling system is presented in Section IV. Experimental results are in Section V and conclusions are in Section VII.

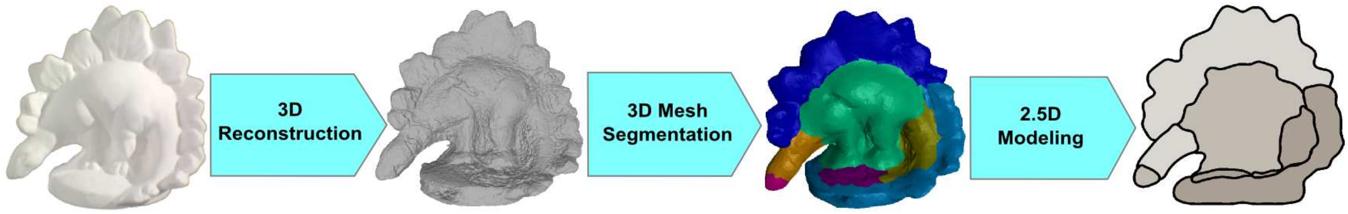


Fig. 2. The overall system framework.

II. Related Work

Many games, animations and applications utilize techniques for simulating 3D scenes using 2D elements. A large number of 2.5D games have emerged during the last decade, including Isometric-Tiles based games such as “Diablo II”, and Parallax-Scrolling games such as “Super Mario”. Most 2.5D techniques fix the viewing angle to a single position, i.e. they can only simulate a 3D scene at a fixed orientation and lack the ability to rotate and view the scene from any angle in 3D. This technique may be traced back to the 1800’s [5].

Some recent applications provide 3D rotation for 2D elements. Di Fiore et al. [4] proposed an approach for animation production that generates key frames by interpolating hand-drawn views. Bourguinon et al. [2] presented another approach using 2D strokes manually drawn on 3D planes. These two methods give the artists less freedom than the 2.5D Cartoon Models recently presented by Rivers et al. [10]. The later is a novel approach for rendering 2D cartoons in 3D. Rivers’ 2.5D model, though, is created purely by manual means. We will take a deeper look at it in Section III. In this paper, we aim to provide an automatic 2.5D modelling technique based on this model.

Recent methods for image-based 3D reconstruction provide better performance and are easier to apply. Using these, one can produce good quality 3D models from photographs taken using hand-held cameras. For the Middlebury Benchmark [11], the most accurate method proposed so far is by Furukawa et al. [6]. The approach of Heip et al. [9] also has good accuracy. We utilize image-based 3D reconstruction approaches to acquire 3D information.

The 3D mesh segmentation benchmark [3] evaluates several approaches to 3D mesh segmentation based on comparisons of human-generated and computer-generated segmentations. The evaluated methods include Randomized Cuts [7], Fitting Primitives [1] and Shape Diameter Function [12]. This comparison allows us to select the most appropriate segmentation approach for our system.

III. The 2.5D Cartoon Model

We now briefly describe the 2.5D cartoon model [10]. Normally a 2.5D cartoon model contains several billboards, each of which is defined in one or more views using one stroke per view, and also has one 3D anchor position, as shown in Fig. 3. The shape of the billboard in a new view is then determined by doing simple 2D interpolation between the corresponding user drawn strokes in existing views.

A. Advantages

The main advantage of Rivers’ model over 3D meshes is that it preserves geometrical cartoon stylistic elements, and not just the appearance style, as provided by Toon-Shading. For example, no matter the viewing angle, Bugs Bunny’s ears are always facing the camera [10]. To present this geometric element in a 2.5D cartoon model, one just draws the ears as they face the camera in all views. This advantage is achieved by defining the shape of the model in each view separately, and editing a shape in one view will not affect that shape’s appearance in other user drawn views.

Another advantage is that downgrading an object from 3D to 2.5D also reduces the data size needed to define that object. For example if a 2D point is defined by 2 variables and a 3D vertex is defined by 3 variables, the data size of a 2.5D rabbit model Ruby, as shown in Figure 4, is only 6% as big as the 3D one. The difference may be even larger as we have ignored the link infomation, which 2.5D points usually have 1-2 links and 3D vertexes often have more than 2 edges.

B. Limitations

There are several limitations to Rivers’ 2.5D models. The one that cause the most issues is the simple interpolation technique used to determine stroke shapes in views that are not key views. The simple interpolation may lead to strange shapes in interpolated views, which may also cause different parts of a model to dettach from each other during rotation. At this moment the only way to solve this problem is to define more key views. In Rivers’ work, three more key views were defined to make the Alien’s arm looks right, as shown in Figure 5. The more views need to be defined the more work artists have to do. While using our system the artist can simply add another

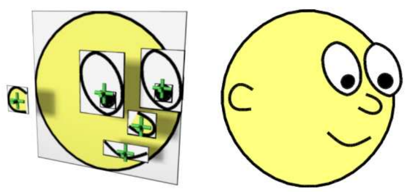


Fig. 3. A picture demonstrating the principle of the River’s 2.5D cartoon models. This simple head model is constructed using 9 billboards, each billboard contains one stroke as its boundary line, a 3D anchor position and a filling colour.

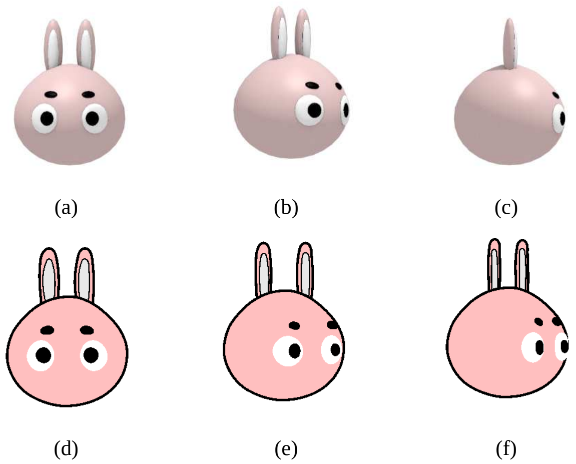


Fig. 4. The front, 45 degree and right view of 3D model and 2.5D model of 'Ruby'. The 3D model is rendered using Toon-Shading. In order to show the advantages of Rivers' structure the 2.5D Ruby has been manually edited based on an automatically generated sketch, we changed the position of eyes and ears in a side view. The eyes can still be seen even when facing 90 degree side, and the ears appear always facing the camera.

line to the commands to generate a new 3D consistent key view, without any extra work.

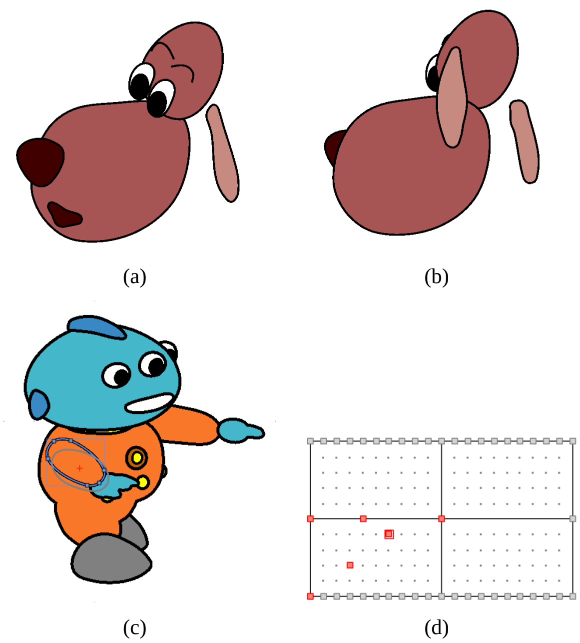


Fig. 5. Notice the ears of the 'Dog' model in Rivers' work [10]. Three more key views are pre-defined to make the arm of the 'Alien' looks right.

C. The manual creation process

Normally the user starts from one view position, draws the main part of the cartoon object, changes to another view position and draws the same parts again, then keeps changing to different views as more parts are added to the model. This process may cost an experienced artist tens of minutes to create a little bear, for example [10]. Thus a method to automatically generate 2.5D models would be very beneficial.

The goal of this paper is to automatically generate 2.5D cartoon models. Rivers' 2.5D cartoon model relies on artists during the drawing process to:

i) analyse the spatial and spectral characteristics of the object
ii) segment the object into parts based on those characteristics, and
iii) create a 2D stroke for each part in each view.

Therefore, to automate the 2.5D model construction, we need to simulate the drawing process of human artists by automating each of the steps described above.

IV. Automatic 2.5D Modelling System

Our approach to automatic 2.5D model creation consists of several steps. A 3D model of the desired object should already exist, or constructed from images using Image-Based Modelling methods such as [6]. Segment 3D mesh into several parts using a 3D mesh segmentation method. The segments are then converted into strokes for creating the 2.5D model. The following sections discuss each step.

A. 3D Information Collection

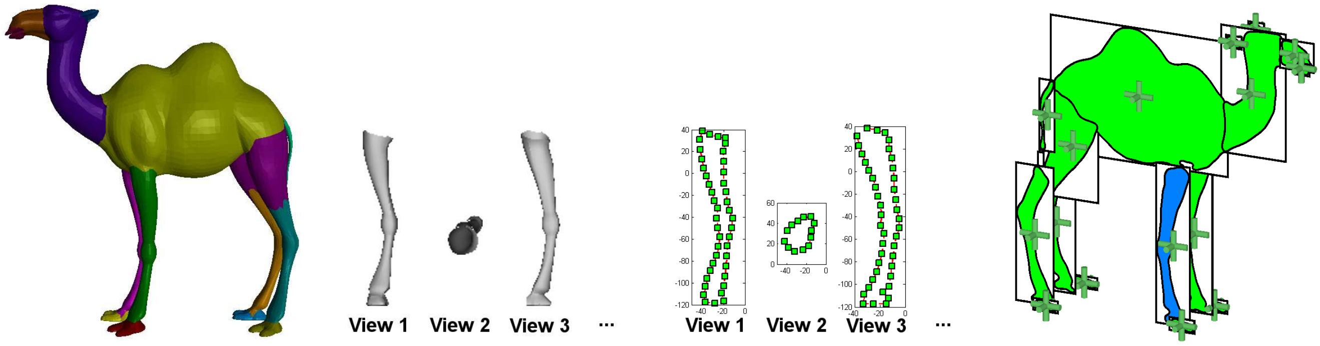
3D information is important for 2.5D modelling, since human artists often use spatial information to segment objects when they create strokes. Also, this information is required to provide 2D object boundaries for creating strokes in any required view. To gather this information, we can construct a 3D model of the object using image-based 3D reconstruction methods. Plausible options include open source software PMVS [6] and Bundler [13], or ready models built manually before.

B. Segmentation

Once we have constructed a 3D model for an object, we can then simulate manual segmentation based on the 3D information. To this end, the segmentation approach should mimic the segmentation by artists and provide similar results. We investigate a set of 3D mesh segmentation techniques and utilize SDF [7], one of the best methods that produces results most similar to those of humans. The actual algorithm used in this work will be discussed in Section V.

C. Parts Refinement

After the 3D mesh segmentation, we cannot yet expect parts are ready to use, because cutting off a 3D mesh from another one makes holes in both parts. If there are multiple holes on a part, from some view angles the camera may see the background through the model. This is not a very rare case, for example if two arms are cut off, in side views the chest will be seen through. The strokes build by the parts having a hole on them will lead to wrong billboard shapes. To solve this problem, we refine parts using a simple 3D hole filling method, which chooses each time the best pair of adjacent border edges and adds a face between them. Several



(a) (b) (c) (d)

Fig. 6. The process of billboard reconstruction: (a)segmented mesh. (b)the right-front leg part from different views. (c)2D control points of the part. (d)final 2.5D model, with the right-front leg marked out.

implementations are available, such as the remeshing filter in MeshLab.

D. 2.5D Model Construction

Once we have segmented the object, we can then convert the segments into strokes to compose 2.5D models in Rivers' framework. Specifically, we create 2D boundaries of each part in each view and then convert the boundaries to strokes.

Firstly, we capture the silhouette of a part in a view by projecting the part into an plane that is normal to the view direction. Once the silhouettes are captured, they are converted into strokes as follows. For each silhouette captured, points on the silhouette are sampled, each of which has a 2D position. These points are rearranged to preserve the right order according to their neighbourhood relationships and linked into strokes. We then create 2.5D models using these strokes.

The 2.5D model construction process is demonstrated in Fig. 6.

E. Summary of the Method

The proposed framework for 2.5 cartoon modelling is summarised in Algorithm 1. The algorithm takes a set of images as input, and the output is a 2.5D cartoon model consisting of a set of billboards.

Algorithm 1 CartoonModelling(I)

Initialise $B \leftarrow \emptyset$, $P \leftarrow \emptyset$
$M \leftarrow ImageBasedReconstruction(I)$
$S \leftarrow MeshSegmentation(M)$
for all $s \in S$ **do**
  for all $v \in Views$ **do**
    $c \leftarrow CreateContour(s, v)$
    $p \leftarrow SamplePoints(c)$
    $P \leftarrow P \cup p$
  end for
  $b \leftarrow CreateBillboard(P)$
  $B \leftarrow B \cup b$
end for
return B

V. Implementation and Experiments

As described in Section IV, our approach requires 3D information of objects. In this experiment we used existing 3D models provided by Strecha et al. [14] and Chen et al. [3].

Our implementation of 3D mesh segmentation is based on Shape Diameter Function [7] since it has good performance for simulating 3D object segmentation by humans (ranked 2 in Chen's benchmark [3]) and is open sourced.

Rivers et al. parameterised the view angle space to a 2D grid containing 440 viewpoints. We normally choose the front, left and up views to construct 2.5D models.

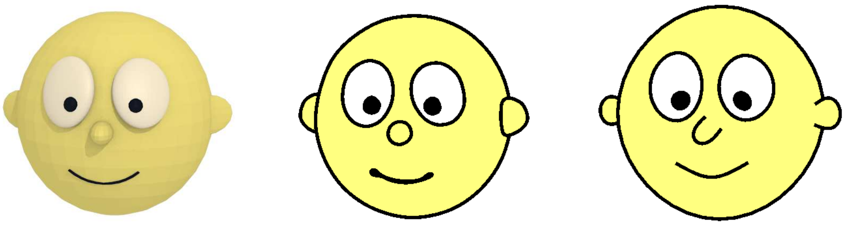
We compare models built by our system with models manually created by humans. To make the comparison more convincing, we select the models originally provided by Rivers et al. [10]. As shown in Fig. 7, our system successfully created models that are close to the fully manually created ones. Comparison of pure 2D and 2.5D scenes of Angry Birds are shown in Figure.10.

The outputs of our implemented system have the same format as Rivers' system, thus models built by our system are ready to be loaded directly into Rivers' drawing software. This makes it very easy to do further manual modifications.

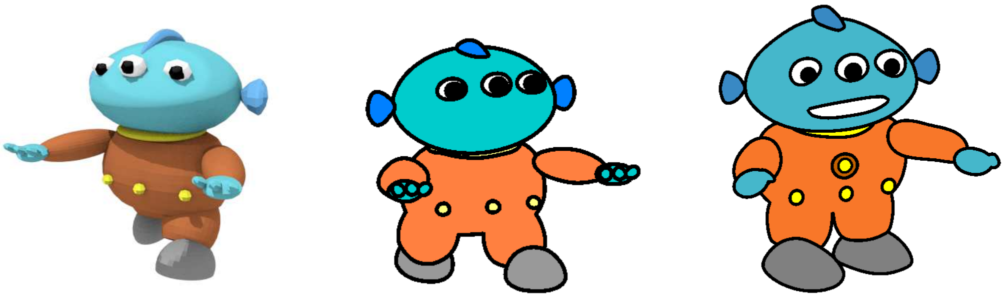
VI. Discussion

Rivers' structure is good for presenting abstract cartoons, but not suitable for models which have long thin shapes, such as tables and chairs. This is because of the limitations of Rivers' structure as discussed in III-B. Complex objects that are not combined by round shapes, normally need more key views to make them look right during rotation. Having many key views makes a 2.5D cartoon model inefficient to render. So converting complex 3D models to 2.5D may not be practically useful in such cases. Therefore, we assume the input models of our system are in the first place suitable to Rivers' structure.

Our system is a general system, which means the system do not take the a cigar and the fingers holding it as different kinds of objects. In some cases there are different ways an



(a) Simple Head, 3D model (b) Automatic by our system (c) Manually by Rivers et. al.



(d) Alien, 3D model (e) Alien, Us (f) Alien, Rivers et. al.

Fig. 7. Some 3D models, example 2.5D cartoon models built using our system (colours are menually adjusted), and manually created 2.5D models by Rivers et. al. [10] are shown for comparison.

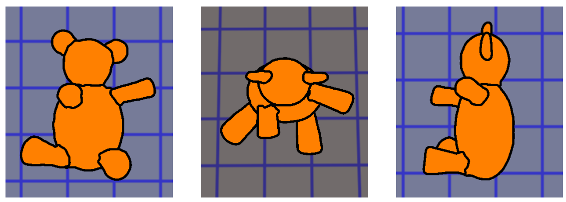
Fig. 8. In some cases, whether the nose stroke is open or closed depends on what style the artist wants to present, picture from [8].

artist might use to present a 3D shape, for example as in Figure.8. In other cases, when a shape obviously should be left open, such as fingers on palm, our system do not automatically keep that stroke open. This part is left to users for now. Future improvements to help the system learn different kinds of objects is a possible direction of research.

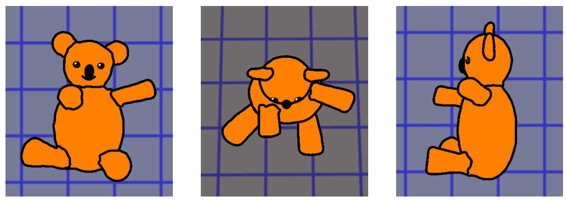
Once a sketch has been built by the system, it could be very easy to add new features on the sketch, as shown in Figure. 9.

VII. Conclusion and Future Work

In this paper, we have presented a novel approach for automatic 2.5D modelling. Our system is able to produce 2.5D models from real-world objects through 3D meshes with minimum human monitoring. It is helpful to artists who want to preserve stylistic 2D elements provided by the 2.5D cartoon model and reduce manual labour. To the best of our knowledge, there is no existing work on automatic 2.5D cartoon modelling, and our approach is the first solution to this problem.



(a) (b) (c)

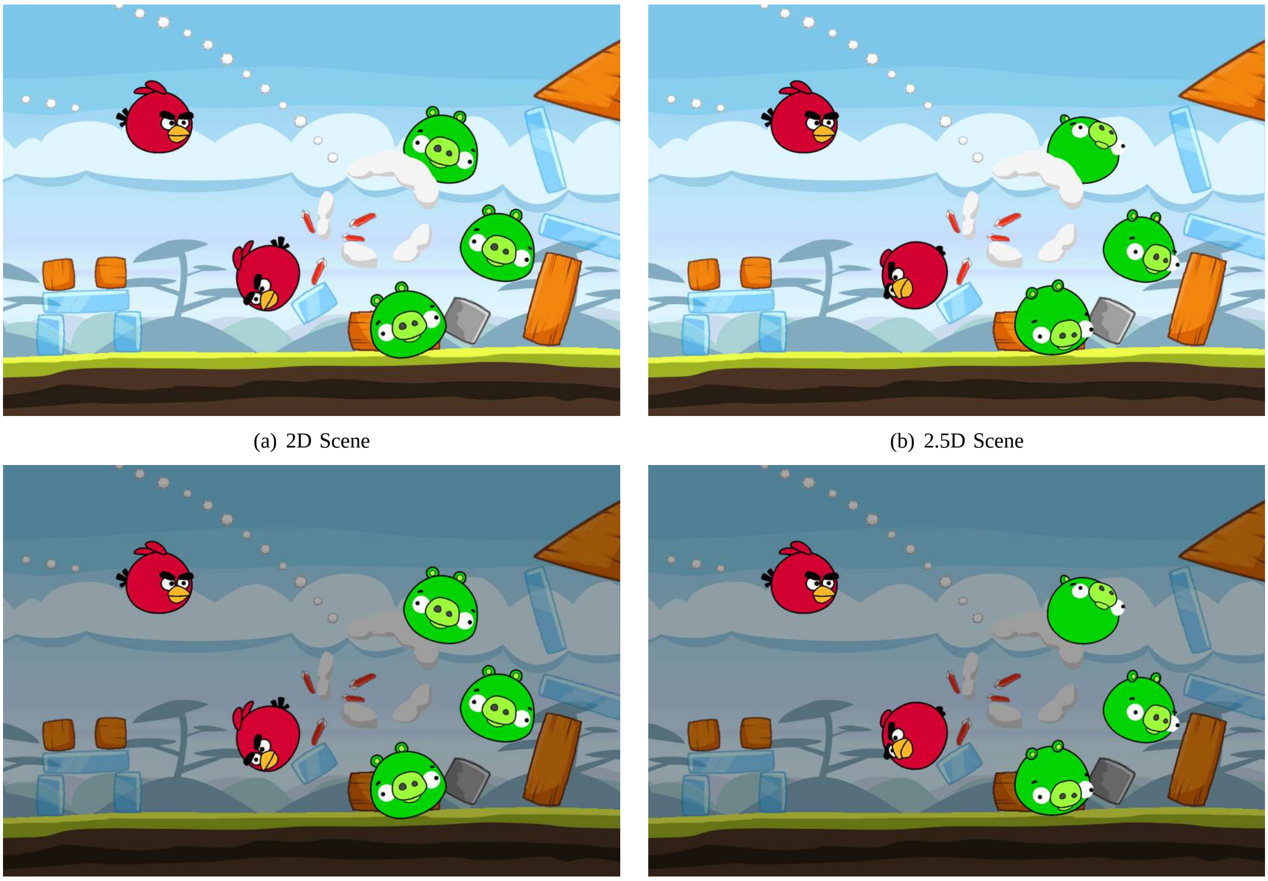


(d) (e) (f)

Fig. 9. Top: A bear model created by our system. Bottom: With only a few manual changes to the automatically built bear, a koala model is created.

Though able to create fully functional models, our system should be considered as a sketch system, that saves artists from basic labour intensive work and lets them focus on the more creative parts, such as colour and style.

In future, we plan to implement a more practical software and improve the billboard reconstruction algorithm to gain better results.



(a) 2D Scene

(b) 2.5D Scene

(c) 2D elements highlighted

(d) 2.5D elements highlighted

Fig. 10. 2D and 2.5D scene of Angry Birds, with comparison of pure 2D and 2.5D elements in scene. 2.5D cartoon birds built using our system.

ACKNOWLEDGEMENTS

We thank Rivio Licensing Team (rivio.com), for granting us the rights to use and create Angry Birds contents for this research. Thanks also to Shy Shalom of intigua for patiently answering our questions about SDF.

REFERENCES

[1] M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, 2006.

[2] D. Bourguignon, M.P. Cani, and G. Drettakis. Drawing for illustration and annotation in 3d. In *Computer Graphics Forum*, volume 20, pages 114–123. Wiley Online Library, 2001.

[3] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3d mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), aug 2009.

[4] F. Di Fiore, P. Schaeken, K. Elens, and F. Van Reeth. Automatic in-betweening in computer assisted animation by exploiting 2.5 d modelling techniques. In *Computer Animation*, pages 192–200. IEEE, 2001.

[5] William Farish. On isometrical perspective. In *Cambridge Philosophical Transactions*. Cambridge Philosophical Society, 1822.

[6] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on Pattern Analysis and Machine Intelligence*, pages 1362–1376, 2009.

[7] A. Golovinskiy and T. Funkhouser. Randomized cuts for 3d mesh analysis. In *ACM Transactions on Graphics*, volume 27, page 145. ACM, 2008.

[8] Harry Hamernik. *Cartoon 360: Secrets to Drawing Cartoon People*. IMPACT, 2010.

[9] VH Hiep, R. Keriven, P. Labatut, and J.P. Pons. Towards high-resolution large-scale multi-view stereo. In *Computer Vision and Pattern Recognition*, pages 1430–1437. Ieee, 2009.

[10] A. Rivers, T. Igarashi, and F. Durand. 2.5d cartoon models. *ACM Transactions on Graphics*, 29(4), 2010.

[11] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *Computer Vision and Pattern Recognition*, 2006.

[12] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259, 2008.

[13] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM Transactions on Graphics*, volume 25, pages 835–846. ACM, 2006.

[14] C. Strecha, R. Fransens, and L. Van Gool. Combined depth and outlier estimation in multi-view stereo. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 2, pages 2394–2401. IEEE, 2006.