Department of Creative Informatics
Graduate School of Information Science and Technology
THE UNIVERSITY OF TOKYO

Master's Thesis

Predicting 3D Bird Flock Motion From 2D Projection

2D プロジェクションによる鳥類の 3D モーションを予測する方法

Che-An Lin
林哲安

Supervisor: Professor Takeo Igarashi

January 2019

Abstract

Bird flock animation plays an important role in making realistic outdoor scenes in games and movies. However, since making bird flock animation involves a large quantity of moving entities, it takes effort to create and control them. To solve this problem, we introduce a new approach to make this task easier. Our main idea is using a single-view RGB video as input. From the input video, by using animal tracking technique, the two-dimensional position of each bird is retrieved in each frame. We then predict the depth information of each bird by minimizing error function based on trajectory smoothness and flock behavior. With the predicted depth, the three-dimensional bird flock animation is then generated. In this research, we focus on synthesizing three-dimensional flock motion from two-dimensional projection. Generating visually plausible result by the system is our goal.

概要

和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．和文概要の文章である．

和文概要の文章である．

和文概要の文章である．

Contents

Chapter 1

Introduction

The art of flock simulation has received increasing interests in the multi-media and entertainment industry. Bird flock scene is widely used in games and movies to make the scenes more rich or natural. However, since the number of objects involved is large, creating bird flock takes considerable time and efforts.

In this paper, we propose an alternative method for modeling bird flock. Our method uses a single RGB bird flock video as input to synthesize flock motion. The goal of our method is not to reproduce the exactly the same flock motion as it in the video. Instead, our method uses video as a reference to synthesize flocks that looks similar and natural to those in the video. We think perfect reconstruction of bird flock is impossible without depth information, as we do not have precise three-dimension motion data as ground truth for evaluating our result.

The work of Raynolds [1, 2] is our starting point about the study of a distributed behavior model. Reynolds proposed boid model, which described the behavior of large groups of birds, herds, and fish with perceptual skills existing in the real world. Inspired by this rule-based method and based on the bird flock behavior we observed, we believe that the motion is predictable from two-dimensional video.

The biggest challenge of this approach is that RGB video does not contain depth information. This makes flock motion synthesizing a difficult task. The reason why we use RGB video as input is, obtaining depth information of bird flock with depth camera is difficult in outdoor scene. For recording locomotion of a single bird as training data, Ju et al.[3] recorded the motion of dove using marker-based optical motion capture and high-speed video cameras for their work. They used 28 cameras to capture a single dove motion in region of 10m × 10m × 7m. Apparently, more space and devices are needed if we do the capture on a bird flock. In fact, we did capture bird flock motion by taking bird video using a 360-degree camera in outdoor scenes, but we failed to get useful video due to bad condition of outdoor environment. And it is also difficult to set up an indoor scene as Ju et al. did for capturing bird flocks, since the space needed is too large. Thus, we consider predicting depth information of bird flock is helpful for making bird flock motion using real video as reference.

Two-dimensional projection data is the key information to predict three-dimension position. The main contribution of this research is to predict three-dimensional from two-dimensional tracking data retrieved from input video. That is, to predict distance from each bird to the camera in each frame. With predefined camera parameters, if we can predict the distance between the bird and the camera, three-dimensional position of the bird can be obtained.

Human eyes can easily track moving objects, but tracking moving objects, especially bird flocks, is a challenging task in computer vision. We use use animal tracking technique for this task. This part is done using interactive feature tracking technique proposed

by Buchanan et al.[4]. Interactive feature tracking is the process of extracting long and accurate tracks of three-dimensional features observed in two-dimensional video. Although the system is designed to track feature points, such as human face or eyes, it is also suitable for tracking bird flock, which contains multiple small trace targets that can also be treated as feature points.

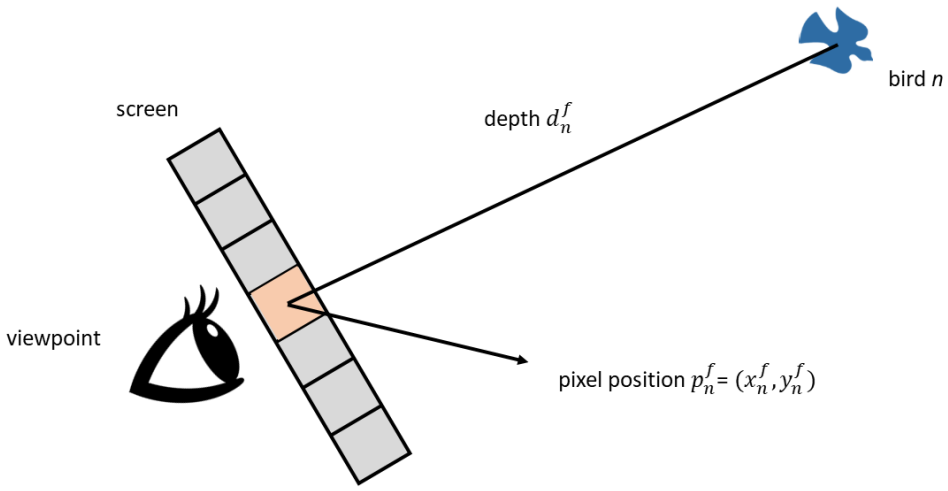

Fig. 1.1. View scenario

After the track data is obtained, predicting distance is next step. Based on our observations and boid flock models, we define two properties that a should have to retain the quality of flock motion. First, considering one bird flying trajectory, we do not expect a bird in frame f to appear too far from its position in frame f-1 from its previous speed. That is, changes in position tend to be gradual. This is the first property: trajectory smoothness. Now we considering a group of birds. Reynolds proposed three steering behavior rules: separation, cohesion and alignment[1]. In case of flock simulation, three rules are considered as three forces applied to each bird in each simulation steps. We also apply these rules in our system as the second property, flock behavior similarity. Considering these rules, target position in frame f can be predicted if we have position in frame $f-1$. We consider an image sequence of length F frames as input, with N birds in each frame. Track data contains two-dimensional projection of each bird in the input video. The set of track data is denoted $P = \{p_n^f\}_{n=1\ldots N, f=1\ldots F}$, where $p_n^f = (x_n^f, y_n^f)$, representing two-dimensional projection on screen, as shown in figure (1.1). We want to predict depth d_n^f of bird n in frame f, while maintain the two properties above. The quality of a flight with a candidate depth set $X = \{d_n^f\}_{n=1\ldots N, f=1\ldots F}$ can be defined as:

$$E(X) = \sum_{n=1}^{N} \sum_{f=1}^{F} e(n, f) \tag{1.1}$$

$$e(n, f) = \lambda_t t(p_n^f, p_n^{f-1}, d_n^f) + \lambda_f f(p_n^f, p_n^{f-1}, d_n^f) \tag{1.2}$$

$e(n, f)$ denotes error function for bird n in frame f, which includes a trajectory smooth-

ness term$t(.)$, and a flock behavior term$f(.)$. λ_s and λ_t are tuning parameters respecting to trajectory smoothness and flock behavior. Thus, the predicting problem can be restated as an optimization problem: choose X to minimize $E(X)$. These terms will be further discussed in detail in later chapters.

Note that our contribution is in predicting three-dimensional flock motion from two-dimensional projection, not in tracking bird flocks.

The system presented in this paper can be separated into four stages. The first stage is trace retrieving stage. In this stage, trace data is retrieved by the system with user indications. Trace data contains projected two-dimensional positions in each frame for each bird. The system only allows user to retrieve trace for only one bird at a time, so trace data must be generated and saved for each bird before going to the next stage.

The second stage is optimization stage. In this stage, optimization is performed based on defined energy function to predict flock motion in three-dimensional space. Since the processing can be done in real time, user can adjust parameters to the systems and see the result directly for better results.

The last stage is refinement stage. Since the system only predict the position in last stage, the orientation of each bird is calculated based on its position in each frame to complete the flock motion as output.

We implemented a flock simulation system to synthesize bird videos that fit the requirements of the system. Despite we used generated results as inputs, we assume the 2D projection to the camera is unknown. This assumption makes the system can also be used for real bird video.

The structure of the paper is as follows: In chapter 2, we discuss related work. In chapter 3, we present the overall design of our system discussing the requirements needed for predicting flock motion. In chapter 4, we present the results generated by our system. In chapter 5, we discuss about future work and limitations of our system. Finally, we summarize this research in chapter 6.

Chapter 2

Related Work

2.1 Animal simulation

Computer graphic researchers have been fascinated by creating life-like computer-generated creatures. Several models are proposed for the nature motion of animals. Most of this research focus on humans or terrestrial animals. Social force model proposed by Helbing and Molnar [5] has been widely used in pedestrian behavior simulation. Miller [6] proposed a physically-based simulation method for snakes and worms. Satoi et al. proposed a unified motion planner [7] that models fish motion with different swimming styles.

2.2 Bird flight simulation

Simulation of bird flight has also gained attention for creating life-like creatures. The work of Wu and Popović [8] describe a physics-based method for synthesis of bird flight animations by given user-specified three-dimensional path. Ju et al. [3] proposed a data-driven approach for controlling flapping flight. These works focus on simulating bird behavior. However, they simulate single bird locomotion based on its structure and aerodynamics, which does not consider the interaction between birds in a flock. Nevertheless, synthesizing realistic behavior of bird is our common goal.

2.3 Crowd simulation

Crowd simulation is the most commonly used simulation system in games and movies to make crowded scenes. Crowdbrush, proposed by Ulicny et al. [9] is a tool for interactive authoring of real-time crowd scenes. As a pioneer of controlling crowds, it provides interface for controlling crowds with brush tools. But the control operations are still limited to small group of individuals by specifying the property or rule, which still needs lots of work for controlling large crowds. Golaem [10] is a mature commercial crowd simulation software widely used in video games. Both tools allow user to manage and control crowds based on path planning and steering behaviors. However, modeling aerial motion is more challenging than humans or terrestrial animals, since birds do not only stay on the ground as they do. Golaem also includes tool for controlling flocks, but it has much less function than tools for crowd simulation, only providing target-based control and collision avoidance.

2.4 2D-based modeling

In computer graphics, Image-based modeling is a method which relies on a set of two-dimension images of a scene to generate a three-dimensional model. Iwasaki et al. [11] proposed a modeling method of clouds from a single photograph. Okabe et al. [12] models volumetric fluid such as smoke or fire, from sparse multi-view images. For most 3D reconstruction like [13], systems are based on shapes from silhouette in multiple views. Generating three-dimensional curves through sketching is also a rich research domain. Pentland and Kuo [14] presented an approach for reconstructing three-dimension object from two-dimensional sketch. In work of Ijiri et al [15], they presented a system for modeling flowers which synthesizes three-dimensional botanical structure by constant curvature with two-dimensional sketch.

Chapter 3

Body

3.1 First

Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body.

Sentences for the body. Sentences for the body.[*1]

Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Table 3.1 shows the result.

3.2 Next

3.2.1 Down

Sentences for the body. Sentences

Table 3.1. The title of this table

col1	col2	col3
itema1	itema2	itema3
itemb1	itemb2	itemb3
itemc1	itemc2	itemc3

[*1] This is a footnote.

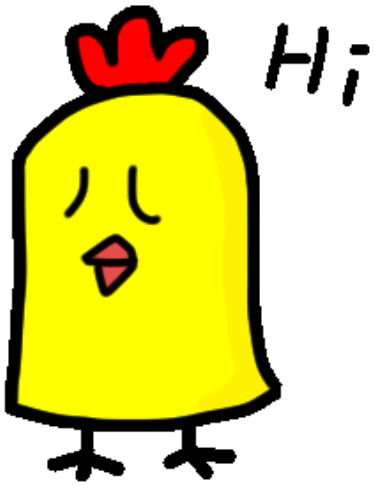

Fig. 3.1. The title of this figure

for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body.

Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body.

3.2.2 Next

Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body.

Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences[*2] for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body.

$$\sum_{k=1}^{n} = \frac{1}{2}n(n+1) \tag{3.1}$$

Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Equation (3.1) implies the conclusion. Refer to Figure 3.1 for details.

[*2] This is a footnote.

3.2.3 Further

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled

with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences. This section shows what the page looks like when filled with sentences.

3.3 Last

Sentences for the body. Sentences for the body.

Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body. Sentences for the body.

Chapter 4

Conclusion

Sentences for Conclusion. Sentences for Conclusion.

Sentences for Conclusion. Sentences for Conclusion. Sentences for Conclusion. Sentences for Conclusion. Sentences for Conclusion. Sentences for Conclusion. Sentences for Conclusion. Sentences for Conclusion.

Sentences for Conclusion. Sentences for Conclusion.

Publications and Research Activities

(1) Sota Akiba. Toward distingushing this and that. Blahblah symposium, Aug 9, 2007.
(2) Sota Akiba, Joichi Hongo. Improvements in analyzing geemie. *Journal of Blahblah*, Vol.1, No.1, pp.1–10, 2007. (in Japanese)
(3) Sota Akiba, Joichi Hongo. Implementation of blahblah. In Procedings of 5th Workshop on Geemie (WOG2008). pp.351–365, 2008.

References

[1] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4):25–34, 1987.

[2] Craig W. Reynolds. Steering behaviors for autonomous characters. *Proceedings of Game Developers Conference*, pages 763–782, 1999.

[3] Eunjung Ju, Jungdam Won, Jehee Lee, Byungkuk Choi, Junyong Noh, and Min Gyu Choi. Data-driven control of flapping flight. *ACM Transactions on Graphics*, 32(5), 2013.

[4] A. Fitzgibbon A. Buchanan. Interactive feature tracking using k-d trees and dynamic programming. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 34(93), 2006.

[5] Peter Molnar Dirk Helbing. Social force model for pedestrian dynamics. *Physical review. E*, 51:4282–4286, 1995.

[6] Gavin S. P. Miller. The motion dynamics of snakes and worms. *ACM SIGGRAPH Computer Graphics*, 22(4):169–173, 1988.

[7] Daiki Satoi, Mikihiro Hagiwara, Akira Uemoto, Hisanao Nakadai, and Junichi Hoshino. Unified motion planner for fishes with various swimming styles. *ACM Transactions on Graphics*, 35(80):169–173, 2016.

[8] Zoran Popović Jia-chi Wu. Realistic modeling of bird flight animations. *ACM Transactions on Graphics*, 22(3), 2003.

[9] Daniel Thalmann Branislav Ulicny, Pablo de Heras Ciechomski. Crowdbrush: interactive authoring of real-time crowd scenes. *SIGGRAPH '05 ACM SIGGRAPH 2005 Courses*, (3), 2004.

[10] Golaem. `http://golaem.com/`.

[11] Makoto Okabe Kei Iwasaki, Yoshinori Dobashi. Example-based synthesis of three-dimensional clouds from photographs. *CGI '17 Proceedings of the Computer Graphics International Conference*, (28), 2017.

[12] Makoto Okabe, Yoshinori Dobashi, Ken Anjyo, and Rikio Onai. Fluid volume modeling from sparse multi-view images by appearance transfer. *ACM Transactions on Graphics*, 34(93), 2015.

[13] Andrew Zisserman Andrew W. Fitzgibbon, Geoff Cross. Automatic 3d model construction for turn-table sequences. *Lecture Notes in Computer Science*, 1506:155–170, 1998.

[14] Jeff Kuo Alexander P. Pentland. Three-dimensional line interpretation via local processing. *Electronic Imaging*, 1249, 1990.

[15] Takashi Ijiriand Shigeru Owada, Makoto Okabe, and Takeo Igarashi. Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints. *ACM Transactions on Graphics (TOG)*, 24(3):720–726, 2005.

Acknowledgements

Sentences used in Acknowledgements. Sentences used in Acknowledgements. Sentences used in Acknowledgements. Sentences used in Acknowledgements. Sentences used in Acknowledgements. Sentences used in Acknowledgements. Sentences used in Acknowledgements.

A

Source code

```
int main () {
  ...
  ...
}
```