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.

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. However, as stated by Reynold, the original boid model can only model flock wandering behavior. Controlling flock remains a necessary task for synthesizing bird flock.

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 nearly impossible without depth information, as we do not have precise three-dimension motion data as ground truth for evaluating our result.

The biggest challenge of this approach is that RGB video does not contain depth information. This makes flock motion synthesizing a difficult task. Human eyes can easily track moving objects, but tracking moving objects, especially bird flocks, is a challenging task in computer vision. The reason why we use RGB video as input is, obtaining depth information of bird flock with depth camera is difficult in outdoor scene. In work from Ju et al.[3], or recording locomotion of a single bird as training data, they 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 they 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 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.

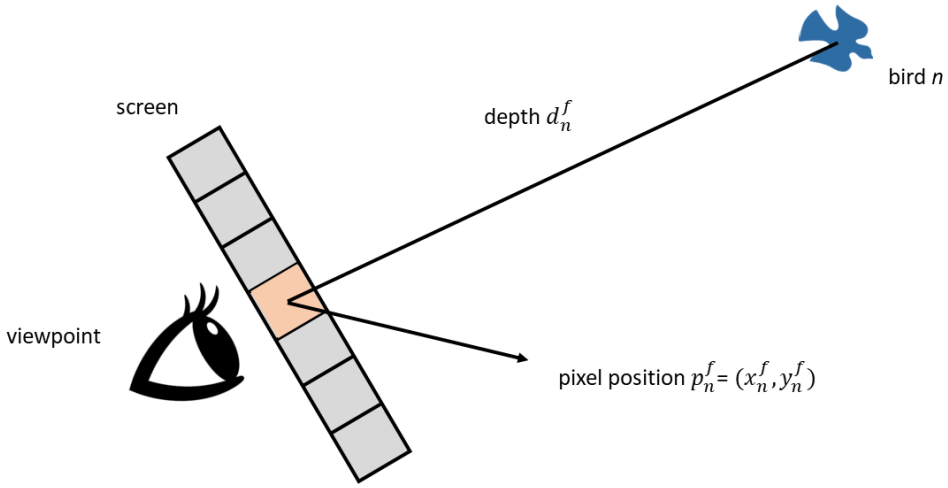

Fig. 1.1. View scenario

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 smoothness 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

Method

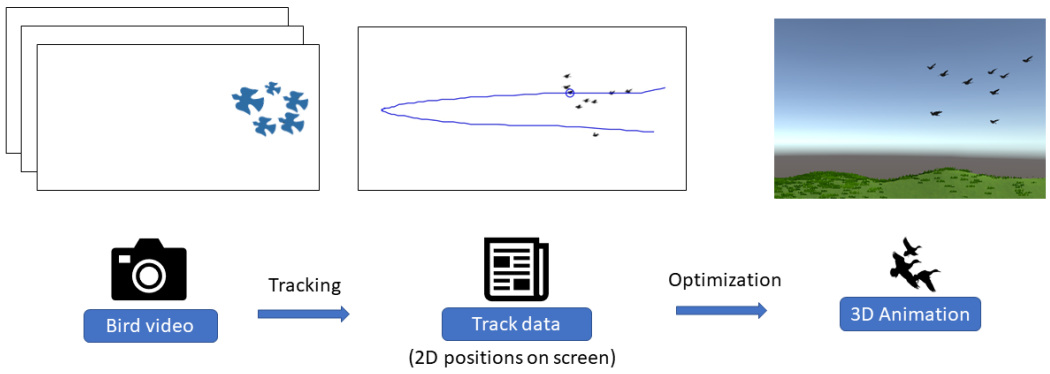

Fig. 3.1. Method overview

3.1 Method overview

Figure3.1 shows an overview of our method. As mentioned before, our method synthesizes three-dimensional bird flock motion from input video. First, trace data of each bird in the video is retrieved. With the interactive feature tracking technique, track data, which contains two-dimensional projection position trajectory of each bird, can be retrieved with the help of user's indications. Next, depth set X is predicted by minimizing error function as Equation 1.1. With the predefined camera parameter and optimal depth d, we can get world position of the bird from its screen position. In most simulation system, velocity of an object is calculated upon mass and force in simulation steps. However, since our optimization method is over frames, we use position-based dynamics system [16], in which velocity in frame f is denoted as

$$\vec{v(f)} = b(f) - b(f-1) \tag{3.1}$$

where $v(f)$ denotes velocity, and $b(f)$ denotes bird position in world space. This makes

our optimization system simpler and faster: we can calculate velocity of each bird in current frame and do not need to keep track of it. Although position-based dynamics system is not physically accurate, since our goal is to make visually plausible flock motion, we use it in our system. For s and t in equation 1.2, we use the simple Euclidean distance function on the two positions. We have also experimented with an acceleration term, of the form $s(p_n^f, p_n^{f-1}, p_n^{f-2})$, but it increases the computation cost, and do not produce better result.

3.2 Bird tracking

To obtain trajectory of each bird in the input video, tracking technique is used in our method. We use tracking system ZooTracer[17], developed by Microsoft research, to do the task. ZooTracer is based on interactive feature tracking technique, proposed by Buchanan and Fitzgibbon [4]. With the tracking system, after preprocessing is done, the user can make tracking data interactively from input video in real time. The system predicts the trajectory of single bird in the video based on user's initial indication. If the tracking is wrong in some frames, user can provide keyframe features by few clicks and improve the track. The main challenge of using interactive feature tracking on tracking birds is distinguishing between tracked birds. The system records fixed-size image patches for tracking feature points, such as eye on a face. However, in case of tracking bird flock, all birds look similar in small patches. It leads to many errors found in the result, making wrong tracking data. Therefore, more manual manipulations need to be done to get an optimal track. On average, user can complete the track data of one bird in a 200-frame video in less than 1 minute.

3.3 Trajectory smoothness

Trajectory smoothness is the first term in our error function in equation 1.1, denoted as $t(.)$. In this term, only the flying trajectory of one bird is considered. We assume that birds minimize its velocity change during flight. The importance of retaining trajectory smoothness is especially obvious when bird is on turning point of the track. In our experiment, we found that if trajectory smooth is not considered, birds will show a sudden turn motion, which is unnatural. Figure 3.2 shows the scenario of sudden turn. Trajectory smoothness term is then denoted as:

$$t(p_n^f, p_n^{f-1}, d_n^f) = v_n^f = b_n^f - b_n^{f-1} \tag{3.2}$$

as error function to be minimized in optimization stage to retain trajectory smoothness.

3.4 Flock behavior similarity

We use Reynolds' boid model as basic flock behaviors. Each bird in flock, which is called boid, observes three steering rules: separation, cohesion, and alignment. The definitions of these steering rules rely on the neighbors of the boid. The neighbors of one boid include its surrounding neighbors who are sufficiently close, and each rule has its perceptual neighborhood. Figure 3.3 illustrates the definition of three rules. To calculate flock similarity, target position of boid n in frame f must be calculated first, and then compared with the predicted boid position. Target position is the sum of the three rules above. Here we further the calculation of three steering rules.

Separation steering behavior gives a boid the ability to maintain a certain separation

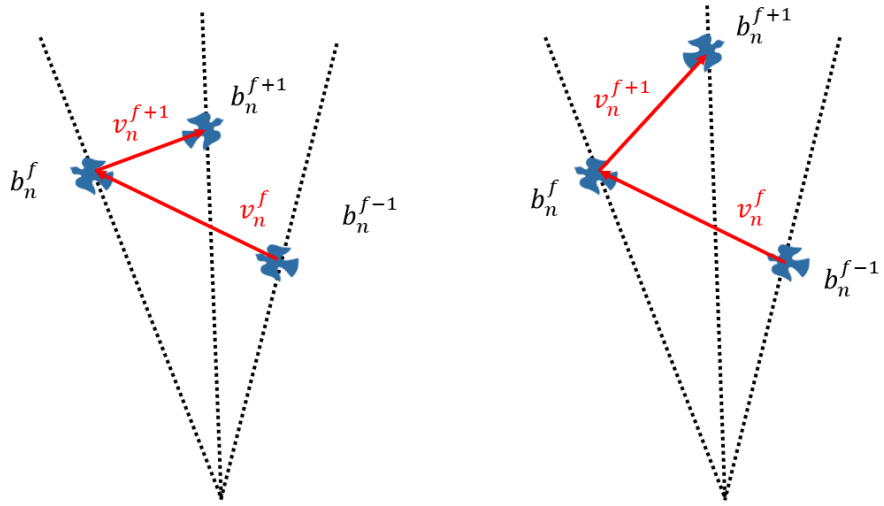

Fig. 3.2. Trajectory smooth. Figure on the left shows the scenario of sudden turn, while the trajectory on the right is considered better trajectory, since the speed difference is smaller.

distance from others nearby and to prevent boids from crowding together. We think that separation rule is the most important rule for creating nature flock motion, since collision with other birds is obviously unnatural. To compute separation force, first a search is made to find other boids within the specified neighborhoods. For each nearby boid, the force is computed by subtracting the positions of the boid and the nearby boids, normalizing, and then applying a $1/r$ weighting, making force of nearer nearby boid stronger. All forces for each nearby boid are summed together to produce the overall separation force.

Cohesion steering behavior give a boid the ability to approach and form a group with other nearby characters. That is, boids steer towards the average position of the nearby boids. This behavior helps flock to form a group and not be too separated. The neighborhood of cohesion is set much bigger than neighborhood of separation does to keep the flock form in group while preventing collision.

Alignment steering behavior gives a boid the ability to align itself, heading in the same direction with other nearby boids. However, since in the optimization stage, we only consider optimizing position of each boid, we do not use alignment rule here. Instead, we refine the alignment after optimization is done.

Combining rules above, now we can calculate flock behavior similarity term $f(.)$ in equation 1.2 by calculating Euclidean distance between the target position and predicted position. Algorithm of calculating flock behavior similarity is shown in figure **??**. (TODO:FIGURE) In most flock simulation approaches, collision is detected in advance by adjusting its direction and speed when other agents are about to collide. In our flock model, since we have separation rule that separate, birds tend to move away to each other when they are too close. Therefore, we do not handle collision avoidance.

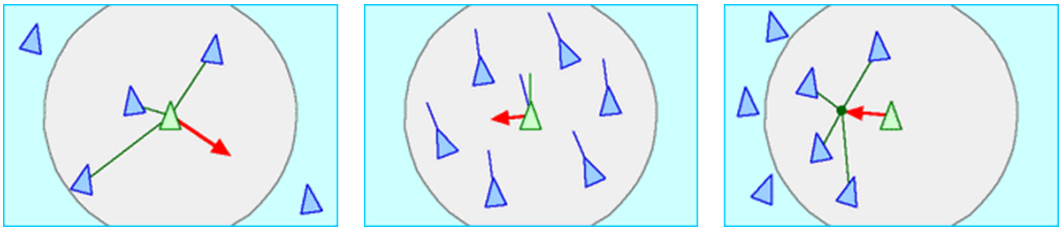

Fig. 3.3. Three boid rules. (left) separation behavior, (middle) Cohesion behavior, (right) alignment behavior.

3.5 Optimization

The task in our optimization stage is to minimize error function equation 1.1 with target depth set X. Totally, since we have track data of N birds in F frames, there are $N \times F$ target depths. If the optimization is done globally, the computation time will be costly. Since our goal is to make system that user can adjust the result interactively in real time, we consider doing the optimization frame-by-frame. Frame-by-frame optimizations means we do not do the optimization in one equation, instead, we predict N depths in frame f from the first frame to next frame, based on the results from last frames. This approach greatly decreases computation time, making it possible for user to do the optimization repeatedly while adjusting parameters. We implement the optimization part using NLopt library [18] with optimization algorithm constrained optimization by linear approximation (COBYLA)[19]. We assume bird flock fly in an area in view field of the camera, a near and a far constraint of depth is set as:

$$d_{near} < d < d_{far} \tag{3.3}$$

Since target error function can be calculated in each frame, with the inequality constraint 3.3, optimized X can be found as result. Algorithm 1 shows our optimization algorithm. (TODO:OPTIMIZATION DETAIL)

By now, we have introduced 5 parameters for optimization stage: λ_t for weight of trajectory smoothness, λ_f for weight of flock behavior similarity, λ_{sep} for weight of separation rule, d_{near} for near constraint of depth and d_{far} for far constraint. In our system, these parameters are treated as interactive elements of our system. Since the optimization can be done in seconds, user can see the result and adjust parameter can do the optimization again to get better results.

Data: Trace set $X = \{d_n^f\}$
// Algorithm start
Assign N bird positions in frame 1 ;
// Frame-by-frame optimization
for $i \leftarrow 2$ **to** F **do**
  $sum \leftarrow 0$;
  // Initial speed
  for $j \leftarrow 1$ **to** N **do**
    $\vec{v} \leftarrow b_j^i - b_j^{i-1}$;
    if $i = 2$ **then**
      $sum \leftarrow sum + ||||\vec{v}|| - \lambda_{speed}||$;
    end
    else
      // Trajectory smoothness
      $\vec{v_{last}} \leftarrow b_j^{i-1} - b_j^{i-2}$;
      $sum \leftarrow sum + \lambda_t||||\vec{v}|| - ||\vec{v_{last}}||||$;
      // Flock behavior similarity
      Calculate separation force $\vec{s}$;
      Calculate separation force $\vec{c}$;
      $target \leftarrow b_j^{i-1} + (\lambda_{sep}\vec{s} + (1 - \lambda_{sep})\vec{c})$;
      $sum \leftarrow sum + \lambda_f||b_j^i - target||$;
    end
  end
  Minimize sum ;
end

Algorithm 1: Optimization algorithm detail.

3.6 Modeling flock motion

After we get the optimized depth data, next step is to visualize the optimized flock motion. We implement it with Unity[20]. And optimization system is loaded as plugin to make it can be used in an interactive way. After the position of each bird in every frame is predicted, to make good flock motion, refinement must be done. Since the system only finds optimal position of each bird, orientation needs to be assigned to complete flock motion. Here we just simply decide orientation of each bird frame-by-frame based on its current position and position in last frame. (TODO:MODELING DETAIL)

Chapter 4

Result

Fig. 4.1. A view of our flock simulation system.

Due to difficulties of obtaining real bird video, we implemented a bird flock simulation system and capture the flock motion as input video to test our system. The simulation system is based on boid model, which assumes a flock is simply the result of the interaction between the behaviors of individual birds. With these behaviors, the system can produce fine flock motion with numbers of birds. However, boid model can only model flock wandering behavior. That is, after assigning initial parameters, all birds become uncontrollable during the simulation, and the simulation always produces same results. To keep the diversity of the generated simulation result, we further include the homing behavior introduced by Xu et al. [21]. With this behavior, we can generate different flock motions efficiently. After the simulation result is generated, we capture videos from it in various angles. Figure 4.1 shows a view of our flock simulation system. The videos are then used as inputs to our system.

We have tested several video inputs generated from flock simulation system. In figure **??**, the generated flock motion contains 5 birds with 200-frames.

(TODO: MORE RESULT)

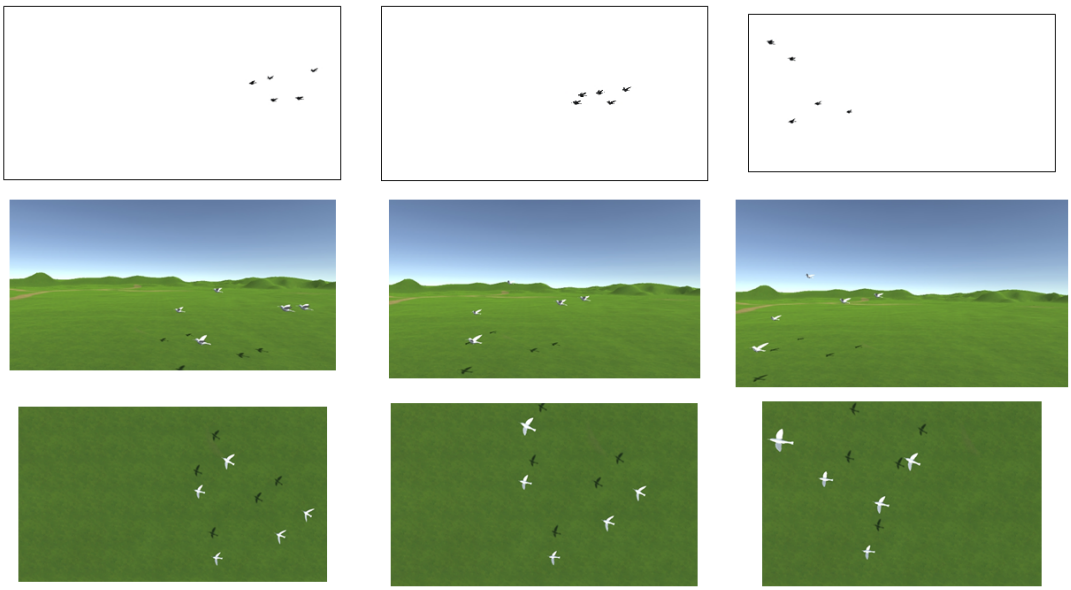

Fig. 4.2. Result1:(top)input video. (middle)generated result from side view. (bottom)generated result from top view.

Chapter 5

Discussion

5.1 Evaluation

In this paper, we aim to synthesize visually plausible flock motion from video by However, as mentioned in Chapter 1, due to difficulties of obtaining data from real flocks, we do not have ground truth to numerically evaluate our method.

5.2 Bird tracking

Another limitation in our system is that we assume all birds stay in the view field of camera. Although it is possible to generate this kind of video from flock simulation system, it is difficult to have such video from real bird flock video. It is common that birds leave and enter the view field, which is not considered in our system.

5.3 Flock modeling

Modeling flocks can be further studied to make better flock motion. In this research we only consider trajectory smoothness and flock behavior. However, we do not consider environmental effect such as wind or obstacles. In addition, bird locomotion is also not considered while it is a critical part for creating natural flock motion. Although the system allows user to change parameters to modify result, the ability of modification is still quite limited.

5.4 Sketch input

The system can be further developed to receive human-drawn sketch as input. Since human sketch is similar to two-dimension track data obtained from video.

(TODO: USER INTERFACE DEATIL)

Chapter 6

Conclusion

In this research, we proposed a novel approach for creating bird flock motion from video based on two-dimensional projections. The proposed system aims to help user to create and model bird flock motion by presenting a bird video as a reference. The system also provides interactive user interface to adjust the generated result in real time by tuning parameters. The benefits of our approach are twofold: providing a new way to create bird flocks by using video input as reference; and the ability to predict three-dimensional flock motion from two-dimensional projection. We tested our system on creating flocks from video, after manual tracking is done, user can adjust parameters and see the result interactively to make better flock motion according to user' s wish. We hope that this research can improve the efficiency of creating rich scenes with flocks in game or movie industry and make it easy for repetitive parameter tuning tasks.

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.

[16] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.

[17] Zootracer. `https://www.microsoft.com/en-us/research/project/zootracer/`.

[18] Nlopt. `https://nlopt.readthedocs.io/`.

[19] M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. *Advances in Optimization and Numerical Analysis*, 275:51–67, 1994.

[20] Unity. `https://unity3d.com/`.

[21] Jiayi Xu, Xiaogang Jin, Yizhou Yu, Tian Shen, and Mingdong Zhou. Shape - constrained flock animation. *Computer Animation and Virtual Worlds*, 19:319–330, 2008.