Realistic Motion Synthesis of Free Falling Behaviours

By Haoran Xie

A thesis submitted to
School of Knowledge Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Knowledge Science
Graduate Program in Knowledge Science

Written under the direction of
Professor Kazunori Miyata

March, 2012

Realistic Motion Synthesis of Free Falling Behaviours

By Haoran Xie (1050024)

A thesis submitted to
School of Knowledge Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Knowledge Science
Graduate Program in Knowledge Science

Written under the direction of
Professor Kazunori Miyata

and approved by
Professor Kazunori Miyata
Professor Kazushi Nishimoto
Professor Katsuhiro Umemoto
Associate Professor Hieu Chi Dam

February, 2012 (Submitted)

Acknowledge

I am eager to express my deep gratitude to my advisor, Prof Kazunori Miyata, for providing me the considerable opportunity throughout my Master course in JAIST. Even through my research topics have transferred among the human skin texture synthesis, snow phenomenon simulation, falling rain, free fall simulation etc., they have been finally settled down to the motion synthesis in physically-based simulation under the supervision of Prof.Miyata. Also thank him for the precious opportunities of participating the important international conferences: NICOGRAPH International and SIGGRAPH Asia, and the research internship at the University of Sydney(Australia) from June to August, 2011. I cannot forget his encouragement when I applied for short-term graduation and IVRC2010 or his wisdom guiding me to be a qualified researcher. Sincerely thank him for accepting me as a research student in Miyata laboratory where I can do my favourite researches and make my dreams come true.

I would like to thank all the members in Miyata laboratory, including Ken Ishibashi, Kaisei Sakurai and Xiaodong Du, for providing their friendly helps and enjoy the research environment. A Special thank you goes to the exchange student Vincent de Paul ATANGANA EKANI who worked with me and completed our first paper about virtual reality in the summer of 2010.

Very special thanks goes also to all my wonderful friends, especial thanks to my best friends Peisheng Hua, Hangyu Li and Chunlei Zhang, who make my research life much more colourful and joyful in JAIST. Also I am very grateful to my German friend in Sydney Robin Pricken who motivated me to cultivate the love of jogging.

Finally, I would like to thank the members in ViSLAB of the University of Sydney. The wonderful conversations with My Advisor Prof Masa Takatsuka and Dr.John Stavrakakis about many research topics stimulated my thought deeply and they support me to complete the paper submitted to SIGGRAPH Asia 2011.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Not all objects fall downward in a straight direction, a piece of paper or a leaf wavers and flutters down in a seeming unpredictable motion when released from your hand. As far as we know, the behaviour of a freely falling object in a vacuity apparatus, or heavy object such as an apple falling from tree is easily and completely understood according to classic physics. To the author's knowledge, the common and spectacular freely falling principle have not been completely resolved in physics, even this researches date back at least to Isaac Newton[Newton, 1687] and James Maxwell[Maxwell, 1854].

Computer graphics researchers have a great ambition to mimic natural phenomena by designing computer algorithms, including rendering, texture mapping, shading, animation technologies, etc.. Recently physically-based simulations have been greatly developed, including fluid simulation, rigid-body simulation and hair simulation, but the author found that the familiar free fall simulation has been ignored and considered as an impossible work due to its intricately inter-wind dynamics in computer graphics. In daily-life, everyone may notice that the free fall motion, sometimes looks like regular, sometimes is so random that we cannot perceive the rules hidden in the particular phenomena. In the common rigid-body simulation and existed game engines, an object is tackled by falling in straight downward direction.

The complexity of free fall simulation lays in the coupling of forward motion to lateral oscillations by surrounding fluid, and the productions and influences of the vortices around the object. The problem of free falling motion involves multiple hydrodynamic effects, including lift force, drag force and vortex shedding, exhibiting both regular and chaotic behaviours. It is a challenging work that relevant to unsteady dynamics, meteorology, flight aerodynamics, sedimentology, bubble rising and boiling, seed dispersal, and all related visual simulations in computer graphics.

(a) (b)

Figure 1.1: Free fall motions of freely falling leaves from tree (from YouTube video) and falling flower petals (a snapshot of an on-line game scene)

1.1 Purpose and objective

Free fall motions (Figure 1.1), such as fluttering (oscillate from side to side) and tumbling (rotate and drift sideways) for lightweight object or strong resistance of high Reynolds number fluid, are spectacular and familiar but we lack physically based approach to synthesize them in computer graphics and virtual reality applications. However, modelling free fall motion is a extremely challenging work.

- First, any locomotion in air flow or other environments with high Reynolds numbers represents a significant simulation and control challenge. The interaction between object and the air is pretty complex. The forces generated by the interaction are chaotic and hard to control.
- The motion is often unstable because of high sensitivity: a slight change of initial angle of object makes significant effect into the result and type of falling motion.
- Furthermore, a specific shape of object also affects motion greatly.

In this research, different types of free fall motions are produced and controlled by initial parameters, including initial released angle, size of object. The simulation is developed in low computational cost for real time request, and provides reliable and natural paths of free fall motion by comparing simulation results with different types of motions from video.

There are some traditional methods to obtain realistic motion synthesis: key frame control, physically-based simulation, motion capture. The author considers that all of these approaches are not available in the following reasons:

- Creating motion by key frame control requires numerous efforts and expert ability for animators;
- Even the physically-based simulation can make a reliable result but it only fits for simple models. Until now, there is no numerical model which can resolve chaotic motion and free fall motion in three dimensional space successfully.

- Motion capture is not only expensive to use due to time consuming and difficult to capture the required motion because of the complex dynamic in free fall motions and dependence of specific circulation, but also is not possible to capture motion data using 3D marker and difficult to control the cameras, especially in a chaotic motion case.

In this work, the author presents new approaches combining both experimental and theoretical results about the topic in physics, and completely examines all motion types in free fall motions of light-weight object to create reliable and natural motion trajectories by using motion planning techniques in a real time environment.

1.2 Research background

Free fall simulation has a rich history in fluid mechanics but few realistic simulations have been achieved in computer graphics.

1.2.1 Physical research

> Every one must have observed that when a slip of paper falls through the air, its motion though undecided and wavering at first, sometimes becomes regular.
>
> ——James Clerk Maxwell, 1854

Scientific interest of free fall motion dates back to one and half century ago, James Maxwell noticed the torque by gravity and lift forces when these forces do not act at the same point[Maxwell, 1854], this is the first study of free falling phenomena. After this work, some achievements have been obtained but the problem is still unsolved. Recently some experimental researches and theoretical models have been proposed to analysis the phenomena qualitatively and quantitatively. However, most of the models are based on the two-dimensional or quasi-two-dimensional experimental setup.

Phase Diagram: The phase diagram for freely falling object with fluttering, tumbling, and steady decent is measured [Willmarth et al., 1964]. They find that the motion depends on six physical characteristics: diameter or the width of the object, thickness of the object, density of the object, density of the fluid, kinematic viscosity of the fluid (air, water, etc.) and gravity acceleration. The phase diagram is conducted using Reynolds number and the dimensionless moment of inertia which are calculated by the six quantities. A more detailed phase diagram is proposed by [Field et al., 1997], they discover a transitional chaotic motion regime between the tumbling and fluttering regimes. Most recently, further experiments get other three basic free fall trajectories in three-dimensional environment: zigzag, transitional helix and spiral[Zhong et al., 2011], which are below the periodic fluttering regime in phase diagram.

Experimental Researches: [Tanabe and Kaneko, 1994] builds a simple model of freely falling paper by solving an ordinary differential equations (ODEs) based on Kutta-Joukowski theorem, using two dimensional flow simulation to identify different regimes of free fall motions. Then Berlmonte et al.[Belmonte et al., 1998] observes fluttering and tumbling motions, they discover the fluttering dynamic of a falling object and the transition from fluttering to tumbling motions occurs at a special dimensionless quantity: the Froude number. Most recently, a complete investigation [Razavi, 2010] concerns about the relationship between different parameters which affect the paths of free falling motions of leaves based on more than six thousands three-dimensional experiments.

Numerical Simulations: There are some works about the numerical simulation of free falling motion in two-dimensional condition. Andersen et al. [Andersen et al., 2005] discuss about the direct numerical simulations of two-dimensional Navier-Stokes equation and a fluid force model based on ODEs derived from experiments and simulations. They analysis the transition between tumbling and fluttering and find an oscillation occurs at bifurcation point. Most recently, researchers develop a Fourier pseudo-spectral method to solve the two-dimensional Navier-Stokes equations coupled with equations which govern free fall motion of a object, and the simulation results are different corresponding to Reynolds numbers[Kolomenskiy and Schneider, 2010]. There is still no convincing numerical simulation approach which can explain the chaotic motion and free fall in three dimensional spaces.

1.2.2 Graphical research

> In general, motion capture creates more realism than what can be achieved computationally, either through procedural (rule-based) animations or (physics- and control law-based) animations. In either case, the challenge is to synthesize motions previously unrecorded.
>
> ——Jim Foley, Getting There: The Ten Top Problems Left[Foley, 2000]

The realistic visual simulations of natural phenomena are the principal mission of physically-based simulation, which are also the significant research field in computer graphics. Over the past decade, researchers have considered a free fall motion as object in the dynamic of changing flow field (e.g. wind field) rather than in a quiescent flow.

Fluid Simulation: Fluid simulation is an important part of physically-based simulation since fluids are ubiquitous and spectacular around us. The earliest work can date back to 1990 by Kass and Miller, and a stable fluids framework presented by Jos Stam [Stam, 1999] made it possible to produce a stable presentation. Then Ron Fediw used a level-set based simulation which facilitated fluid simulation become popular, many extensions and improvements were proposed. Fluid simulation is practicable for fire, smoke, sand and particle-based explosions nowadays. In real time application, simulation framework is often using Eulerian fluid simulation approaches based on semi-Lagrangian method.

As other approaches, vortex filaments based simulation is presented by Angelidis which gives artists full control over simulation result, and recently Weimann et al. model moving obstacles with vortex shedding[Weissmann and Pinkall, 2010]. Smoothed Particle Hydrodynamics (SPH) originated from astro-physics was firstly introduced into real time simulation of liquids by Müller[Müller et al., 2003], and later extended to handle multi-phase fluids. Because SPH does not require a global correction step for the incompressibility of fluid, the resulting compressibility artifacts are a research topic for the current fluid simulation [Yu and Turk, 2010]. Lattice Boltzmann Methods (LBM) also become an alternative because of its simplicity and efficiency, it was used for wind simulations with interacting with rigid bodies [Wei et al., 2003], falling snow in real time, and simulations of free surface flows. Another approach is Shallow Water Equations (SWE) which reduced the form of the Navier-Stokes equations, so that can be resolved efficiently because of their two dimensional nature. SWE have been used for simulating splashes or coupled to terrains and rigid bodies. Many effects can be achieved by applying a semi-Lagrangian advection to the SWE with an implicit time integration scheme, such as bubble dynamics and terrain erosion [Mei et al., 2007]. There are a variety of approaches exchanged from Computational Fluid Dynamics (CFD) community in computer graphics like Vortex Methods etc. [Koumoutsakos et al., 2008]. Recent works are popular to combine grid-based and particle-base methods to obtain a good resolution result with lower computation consume [Narain et al., 2010].

Falling Simulation: The research of complete free fall simulation is not so satisfying in computer graphics. In the scarce physically-based simulations, Wei et al. [Wei et al., 2003] used LBM method for wind simulation and rendered soap bubbles and a feather as results, this approach cannot obtain the designated motion trajectories, also it is very heavy for simulating multiple objects due to the particle method consuming computational cost (Bubble: CPU 2.8fps GPU 11.5fps; feather: CPU 0.76fps GPU 6.1fps). Other related example-based approaches were based on Markov model [Reissell and Pai, 2001], captured videos [Aoki et al., 2004], segments of fluid Simulation [Shi et al., 2005], animated trajectory by Maya[Vázquez and Balsa, 2008], sketch examples [Haiyan et al., 2010]. All these simulations ignored the nature of free fall and considered the free fall motion as a complete complex and unpredictable dynamics using stochastic processes or simple particles representation.

Some commercial CG tools, including Lightwave, Maya, etc., are actually loss of the function of free fall animation, uses particle simulation for falling objects by modifying drag and lift force parameters in wind field. In these cases, the motion paths are unpredictable and not easy to be controlled and modified for required motions.

Example-based Motion Synthesis: Example-based or data-driven motion synthesis is to combine the controllability of procedural animation or physically-based animation with realistic appearance of recorded motion stream, such as motion capture. The first

paper about automatically organizing the example motion clips into graphs for efficient motion syntheses proposed a motion graph [Arikan and Forsyth, 2002]. Later Kovar et al. build an extension motion graph using local search with branch and bound algorithm [Kovar et al., 2002]. Comparing with other methods in character animation, the example-based approaches have special advantages: on the one hand, the synthesized motion keeps the visual reality as those in motion database; on the other hand, the motion synthesis process is controllable.

Rather than character animation, motion graph is used in other physical based animation, such as tree animation [Haevre et al., 2006] [Zhang et al., 2007]. These thesis builds upon motion graph in synthesizing a freely falling motion.

Others: Other interesting related works are concerning the oscillations in motion synthesis and unsteady dynamics, including rain streaks [Garg and Nayar, 2006], bubble rising [Hong and Kim, 2003], kite flying [Okamoto et al., 2009]. In the paper [Desbenoit et al., 2006], the authors referred five types primitive free fall motion for calculated the accumulated leaves on ground, but they ignored the realistic trajectory synthesis of free falling motions.

As Jim Foley [Foley, 2000] said about the top ten unsolved problems in computer graphics in 2000, "Fill the gap between motion-capture animation and simulation/procedural animation" was one of the problems. A simulation not only creates more realism but also keep the controllability. For many complex natural phenomena including chaotic motion as free fall, It is urgent to propose the approaches to fill the gap between physically-based simulation and example-based animation. This work is one of the research challenges in computer graphics.

1.3 Contributions

In the physically based simulations of computer graphics, free fall animation is a new topic and useful in many physical simulations, especially, simulations of natural phenomenon. When objects start falling, oscillations happen commonly but often are neglected in the previous researches. This work is a pioneering work to introduce a motion synthesis approach, including motion classification, primitive motions and motion groups. This work adopts a data-driven approach from a set of pre-computed motion databases. In our simulation, fluttering, tumbling, chaotic motion types are defined based on the phase diagram, and also the statistical results based on thousands of experiments and the numerical simulation of two dimensional fluid are connected in this work to present the nature of free fall motions.

Comparing to related work in the research background, this work achieves a more realistic free fall simulation in low computational cost. This is a challenging work which tries to resolve chaotic trajectories and the trajectories in three dimensional by motion synthesis approaches. This framework is compatible with the existed frameworks of rigid-

body simulations. The animator or programmer can automatically synthesize the free fall trajectory in a nature-like or what they wanted.

The key contributions of this thesis are as follows.

Trajectory Search Tree: The author has compared the trajectories of periodic fluttering, tumbling and chaotic motion trajectories from actually measured data, and find that there are two alternatives when the object arrives to a turning point, in the mean time, chaotic motion happened if objects oriented vertically at turning points. In this condition, we mix the motion segments of fluttering, considering that the chaotic trajectories is an unstable and transitional regime between fluttering and tumbling motion.

Pre-computed Trajectory Database: Since the motion of free fall is complex, it needs a plenty of memory cost consuming by using numerical simulations, which is not appropriate for real time applications. Our approach is to calculate the position and orientation off-line and stored into a database of motion segments. This database is used by choosing proper feature vectors from on-line environments. Our database includes about 1200 motion segments including the data of positions and calculated orientations. The orientation is depending on the interpolation of data by solving ordinary differential equations of a force model of free fall. The motion is generated by piecing together example motions from the database.

Free Fall Motion Graph: This work has developed a specific motion graph based on six primitive motions distributed in the phase diagram of free fall. We construct the global motion path by selecting sequence of nodes from different primitive motions using discrete-time Markov chain model. By analysing the motion data of thousands of experiments in previous work, we make a hypothesis about the motion selection which never be presented neither physics nor computer graphic researches: There are 35 kinds of the potential motion types for free fall motion in the case of period spiral phase for an example. Also the probability of motion selection is created to get numerous realistic but differential motion trajectories of free fall.

1.4 Overview and organisation

1.4.1 System overview

The configuration of our system is shown in Figure 1.2. The inputs of our system are the initial conditions and parameters of a lightweight object with six degree of freedoms (DOFs), including the physical characteristics of object and fluid wherein released (release height, mass, etc.). We transform the parameters to two important non-dimensional numbers Re and $I*$, and lookup the phase diagram of free fall to query the motion level in which the primitive motion of the object is stable.

After the motion level of primitive motion is decided, we first define the motion types category based on the level. In all kinds of the motion types of the decided motion category, a discrete time Markov chain model is adopted to choose the actual motion paths. In our work, the motion trajectory is constructed from six motion sequences of primitive motions. The free fall motion graphs are the used to synthesis the trajectories from the designated motion segments. After optimization to the initial trajectories, final fall paths are synthesized. Additionally, the falling motion in wind field is simulated using an improved noise-based wind field approach.

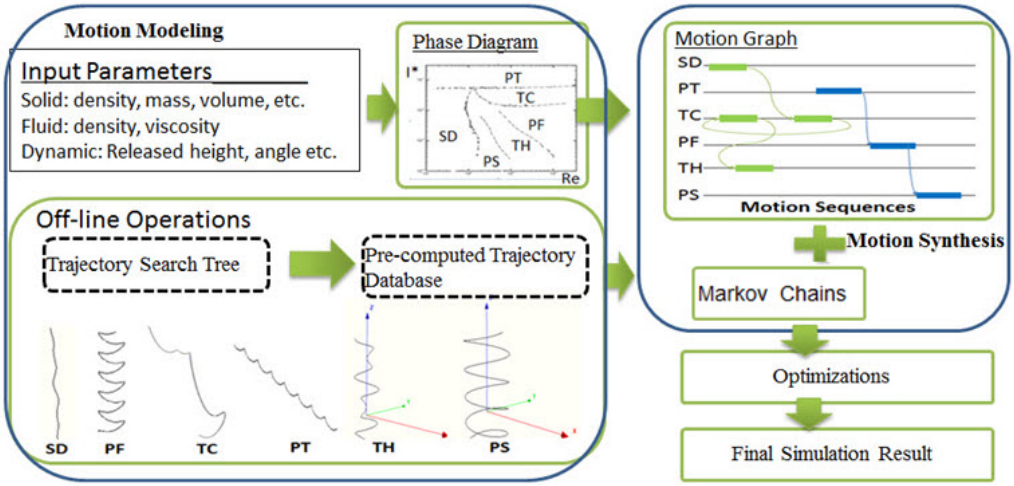


Figure 1.2: System overview

1.4.2 Thesis organisation

The remaining chapters of the thesis is organized as follows:

- Chapter 2 gives the necessary knowledge in physics related to free falling motion, including the analysis of the basic equations based on the Kutta-Joukowski theorem. Then the phase diagram is introduced by important dimensionless quantities. Finally a simple quasi-2D force model is described.

- Chapter 3 demonstrates the motion modelling of the free falling motion and explains how to create free fall trajectory database for synthesizing the individual primitive motions.

- Chapter 4 displays the new free fall motion synthesis approach using motion graph and gives an unprecedented motion classification of free falling motions.

- Chapter 5 explains the free fall motion in wind field, the wind fields are implemented by noise-based method.

- Chapter 6 shows the experiments and simulation results of this work, comparing the simulation results to the ground truth from experiments.
- Chapter 7 discusses the conclusion of this work, and outlines the future research directions.

Chapter 2

Physical Acquisition

The basic knowledge of free fall phenomenon is described in this chapter. In theoretical researches, the fundamental theorems of aerodynamics are simply introduced, and then the two-dimensional ordinary differential equations based on the force model and its improvements are explained. In experimental researches, a phase diagram is introduced into this work through observing numerous experiments in previous works.

2.1 Introduction

Free fall is a complex phenomena that we cannot understand the motion completely until now. In this chapter, the qualitative and quantitative specifications are introduced. As the basic theorem of the fluid and rigid-boy in flow, the Navier-Stokes equations and Kutta-Joukowski theorems are simply explained.

The *Re*-I^* phase diagram based on numerous experiments is significantly important for analysing the motions of specific object. Here, the physical meanings of the two dimensionless quantities, Reynolds number and the dimensionless moment of inertia are given in advance.

Even there still isn't any proper numerical simulation of free falling in three dimension, the simple phenomenological model in 2D or quasi-2D condition is useful and feasible. We distinct the primitive motions of free fall by solving the ODEs. Also, in an extensional model, Frouder number is introduced to explain the transition from fluttering to tumbling motions.

2.2 Navier-Stokes equations

The governing equations for incompressible viscous fluid are given by:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \vec{u} + \vec{g} \tag{2.1}$$

$$\nabla \cdot \vec{u} = 0 \tag{2.2}$$

where ν is the kinematic viscosity of the fluid, ρ is the density of the fluid and $\vec{g}$ is the gravitational acceleration. The equations are called Navier-Stokes equations, which describe the fluid in terms of a continuous velocity field $\vec{u}$ and pressure p. Equation 2.1 dictates the conservation of momentum, Equation 2.2 is the conservation of mass for dynamic of fluid.

There also have a Lagrangian form of the Equation 2.1 using $\frac{D}{Dt} = \frac{\partial}{\partial t} + \vec{u} \cdot \nabla$, so the equation can be substituted by

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \vec{u} + \vec{g} \tag{2.3}$$

The above equation can be written by the Newton's second law form as follows:

$$\vec{g} = \frac{1}{\rho}(f^{pressure} + f^{viscosity} + f^{advection}) \tag{2.4}$$

There are three terms in the right side of the Equation 2.4. The first pressure term involves the gradient of the pressure, the second viscosity term involves the second derivates of the velocity with the viscosity constant accounting for the "thickness" of the fluid(e.g., water has much higher viscosity than air). In computer graphics, the semi-Lagrangian method [Stam, 1999] computes the above equations by different terms separately to make a stable and elegant algorithm.

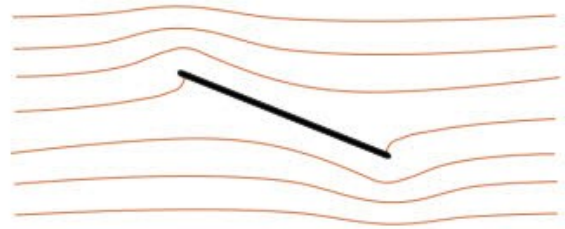
2.3 Kutta-Joukowski theorem

The reason why lift force is generated in flow field, is because of the pressure distribution of the upper and lower surface of an object. According to the Bernoulli's equation [Tritton, 1988],

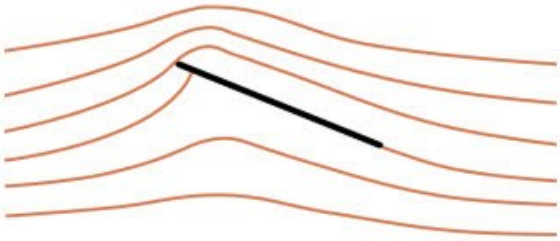
$$\frac{1}{2}\rho q^2 + p = \text{constant along a streamline} \tag{2.5}$$

where ρ and q are the density and magnitude of the velocity ($|\vec{u}|$) of the fluid, p is pressure. From Equation 2.5, the relationship between velocity and pressure is inverse, i.e, the pressure becomes small if the velocity of flow is high and vice versa. In Figure 2.1 (b), above the upper surface of the object the streamlines are pushed together, the fluid moves faster than the free flow speed. It deduces that the pressure of upper surface p_{top} is reduced. Similarly, the pressure at the lower surface p_{bottom} of the object is increased because of low velocity $\vec{u}_{bottom}$. The pressures of two surface are different so that there is a upward force on the object, which is called **Lift** (see Figure 2.2). Because of the vorticity of the natural flow air, the **Circulation** is generated which can explain the higher velocity at the upper surface of object.Let C be the closed curve of the object contour, the circulation is defined as

$$\Gamma = \oint_c \vec{u} \cdot dx \tag{2.6}$$



(a)



(b)

Figure 2.1: Inviscid flow streamlines around object:(a) flow without circulation-unnatural streamlines; (b) flow with circulation-natural streamlines

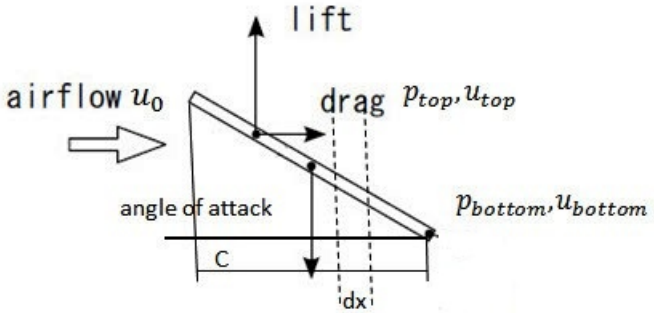


Figure 2.2: The exerted forces on a falling object in flow field

From Bernoulli's equation, it implies that $\gamma \neq 0$. Figure 2.1 (a) notices that the flow without circulation is in the condition $\gamma = 0$, which is unnatural.

The Lift on an object can be calculated as

$$\begin{aligned} L &= \int_0^C (p_{bottom} - p_{top})dx \\ &= \int_0^C \frac{1}{2}\rho(\vec{u}_{top}^2 - \vec{u}_{bottom}^2)dx \\ &= \frac{1}{2}\rho \int_0^C (\vec{u}_{top} - \vec{u}_{bottom})(\vec{u}_{top} + \vec{u}_{bottom})dx \\ &= \rho\vec{u}_0 \int_0^C (\vec{u}_{top} - \vec{u}_{bottom})dx \\ &= -\rho\vec{u}_0(\int_C^0 \vec{u}_{top}dx + \int_0^C \vec{u}_{bottom}dx) \\ &= -\rho\vec{u}_0\Gamma \end{aligned} \tag{2.7}$$

The equation $L = -\rho U\Gamma$ is called the Kutta-Joukowski theorem, where $U = \vec{u}_0$ is the velocity of the flow field. The following equation is also obtained for a symmetrical object (e.g a leaf, a piece of paper).

$$L = l\rho\pi U^2 \sin\alpha \tag{2.8}$$

where l is the length of the object, α is the angle of attack in Figure 2.2. Equation 2.8 is useful for analysing the external forces on a free falling object in this thesis. The numerical simulation model can deduce the ODEs based on this equation, I will explain the model in Section 2.5.

Now the lift force can be calculated by the Equation 2.7, the external forces $\vec{f}$ by combining gravity and lift force can replace $\vec{g}$ in the Navier-Stokes equations (Equations 2.1).Because these equations is based on two-dimensional flow, it is more complicated and unsolved due to turbulence in three dimensional conditions. So far, by solving Navier-Stokes equations using particle-based approach (SPH, Semi-Lagrange Method, etc.), the approach is available but cannot get the realistic free fall trajectories. The work [Okamoto et al., 2009] used a similar approach to simulate the wavering motion of kite in the sky, however, the simulation results are not natural and realistic.

2.4 Free fall phase diagram

A falling object is characterized by the following quantities(See Appendix):

- h: height to release
- L: length of object
- a: length of the cross section of the object
- b: width of the cross section of the object

- ρ_s: density of the object
- ρ_f: density of the fluid
- ν: kinematic viscosity of the fluid
- g: gravity acceleration

Namely, a is the width and b is thickness of the object; In the case of ellipse, L and a are the major and minor axes; If circular, $L = a$ is diameter. From these parameters, three dimensionless quantities: the Reynolds number, Re; the aspect ratio of object, $\epsilon = \frac{b}{a}$; the dimensionless moment of inertia, I^* are calculated. In the subsection 2.4.1 and subsection 2.4.2, the physical characteristics of Reynolds number Re and the dimensionless moment of inertia I^*, are two key quantities for building the phase diagram.

2.4.1 Reynolds number

To obtain the dimensionless form of the Navier-Stokes equations, the variables are given as:

$$u' = \frac{\vec{u}}{U}, t' = \frac{U}{L}t, x' = \frac{1}{L}, p' = \frac{1}{\rho U^2}p \tag{2.9}$$

where U is the velocity scale of the flow. Substituting Equation 2.9 into Equation 2.1 and 2.2 gives

$$\frac{U}{L}\nabla' \cdot u' = 0 \tag{2.10}$$

$$\frac{U^2}{L}\frac{\partial u'}{\partial t'} + \frac{U^2}{L}u' \cdot \nabla' \cdot u' = -\frac{U^2}{L}\Delta p' + \frac{\nu U}{L^2}\nabla'^2 u' \tag{2.11}$$

where $\nabla' = \vec{x}\frac{\partial}{\partial x'} + \vec{y}\frac{\partial}{\partial y'} + \vec{z}\frac{\partial}{\partial z'}$. Hence,

$$\nabla' \cdot u' = 0 \tag{2.12}$$

$$\frac{\partial u'}{\partial t'} + u' \cdot \nabla' \cdot u' = -\Delta p' + \frac{1}{Re}\nabla'^2 u' \tag{2.13}$$

Defining

$$Re = \frac{UL}{\nu} \tag{2.14}$$

Re is a dimensionless quantity called the Reynolds number, which is presented for the flow patterns with different dynamic similarities. In Figure 2.3, for small Reynolds number flow, the inertia of the flow is not important, the flow is smooth and straight forward; For higher Reynolds number, inertia begins to play an important role and vortices are generated behind the object. At more higher Reynolds number, where inertia is dominant, vortex shedding starts, i.e. the vortices are not stationary any more but detach from the top and bottom of the object, which is known as *Von Karman vortex street* (Figure 2.3 (c)). At the highest Re, turbulence arises and the motions of flow behind object are filled

with chaotic motions.

Commonly, the velocity scale U is approximated by the average descent velocity of the falling object.

$$U \sim \sqrt{(\frac{\rho_s}{\rho_f} - 1)gb} \tag{2.15}$$

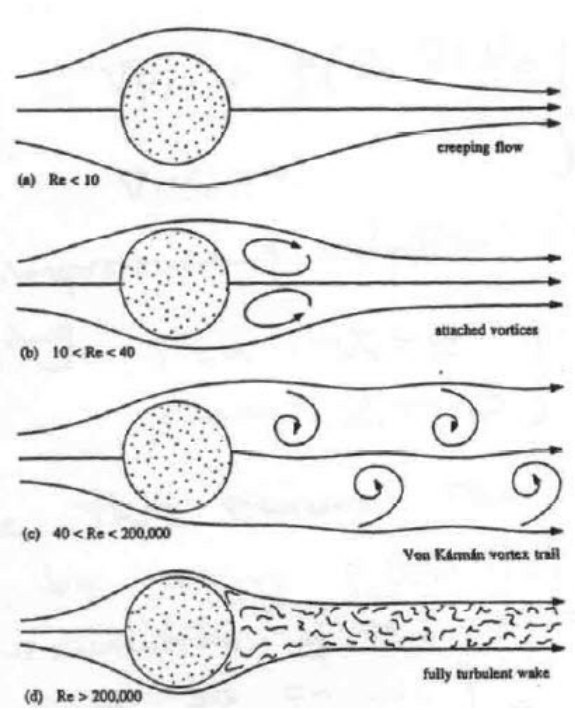


Figure 2.3: The flow pattern around a cylinder of the different Reynolds numbers Re, from [Bruce R. Munson, 2001]

2.4.2 The dimensionless moment of inertia

Willmarth et al.[Willmarth et al., 1964] proves that the dimensionless moment of inertia I^* of the falling object is a significant quantity to indicate the inertial resistance of the object to rotate in free fall motions. I^* is defined as the ratio of the moment of inertia of a thin object and a quantity proportional to the moment of inertia of a rigid sphere of fluid with diameter d.

$$I^* = \frac{I}{\rho_f d^5} \tag{2.16}$$

where I is the moment of inertia of an object, which is calculated by

$$I = \int_V \rho(x,y,z) \begin{bmatrix} y^2+z^2 & -xy & -xz \\ -xy & z^2+x^2 & -yz \\ -xz & -yz & x^2+y^2 \end{bmatrix} dxdydz \tag{2.17}$$

where $\rho(x, y, z)$ is a density function of object.

This work only considers about the simple uniform mass distribution objects, the calculation results of I^* in the common cases of primitive geometries are given as follows

For disks,

$$I^* = \frac{\pi \rho_s b}{64 \rho_f a} \tag{2.18}$$

For plates with the rectangular cross section,

$$I^* = \frac{8 \rho_s (a^2 + b^2) b}{3 \pi \rho_f a^3} \tag{2.19}$$

For plates with the elliptical cross section,

$$I^* = \frac{\rho_s (a^2 + b^2) b}{2 \rho_f a^3} \tag{2.20}$$

In this work, these three cases are used to approximate the falling object geometries, for examples, coin corresponding to the disk case, leaf corresponding to the elliptical case, etc.

2.4.3 Phase diagram

About the three dimensionless quantities Re, I^* and ϵ, ϵ is so small due to the light-weight objects (thin objects) that we omit affect by the aspect ratio of the object($\epsilon \ll 1$).

The Re-I^* phase diagram is introduced by [Willmarth et al., 1964] fifty years ago. To date, the important works by ([Willmarth et al., 1964], [STRINGHAM and GUY, 1969], [Field et al., 1997], [Zhong et al., 2011]) complemented the phase diagram to three dimensional cases (Figure 2.4).

The phase diagram shows the dynamical behaviours of free falling objects as a function of the two dimensionless parameters Re(abscissa) and I^*(ordinate). For increasing Re and I^*, the trajectory becomes more planar, and tumbling motion (see Figure 2.4 (b)) happens entirely in the same 2D plane. For decreasing Re and I^*, the trajectory becomes more circular, and the spiral motion is totally in three dimension (see Figure 2.4 (f)). If $Re < 10^2$, the trajectory becomes steady.

In other words, the free fall motion becomes more sensitive with higher Re and smaller I^*. This phenomena is related to the oscillations happened in multiple directions. When the object starts falling in a flow, planar oscillation occurs firstly, and becomes unstable with smaller I^*. For smaller I^*, a secondary oscillation grows in the normal direction of the original falling plane. For higher Re, the oscillation in planar direction attenuates slowly. The free fall motion is easier to oscillate and becomes unstable in the case of higher Re

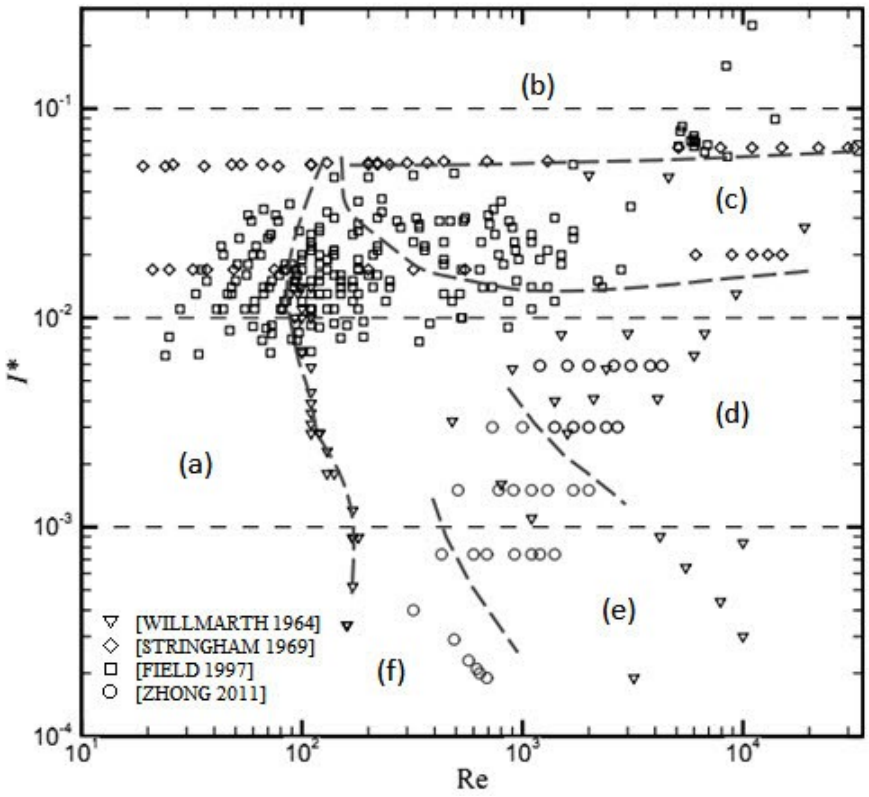


Figure 2.4: The Re-I^* Phase Diagram of free fall motions, including six regimes:(a) steady descent motion, (b) tumbling, (c) chaotic, (d) fluttering, (e) helix and (f) spiral motions. The symbols in the diagram are verified by previous experimental works.

and smaller I^*.

The primitive motions of free falling are illustrated in Figure 2.5 and interpreted as follows.

- **Steady Descent motion (SD)** The object drops down straight forward in vertical direction. As shown in Figure 2.4 (a), SD happens at smaller Re ($Re < 10^2$) regimes.
- **Periodic Tumbling motion (PT)** The object turns continuously end-over-end and drifts in one direction at higher I^* regime in phase diagram.
- **Transitional Chaotic motion (TC)** Chaotic motion is observed when the object oscillates, the amplitude becomes larger and larger, finally turns into a tumbling motion. It is the transitional motion between tumbling and fluttering motions. Actually the regime of TC is at the medium Re and I^*, in the middle regime of the tumbling and fluttering regimes (Figure 2.4 (c)).
- **Periodic Fluttering motion (PF)** Periodic fluttering is found at higher Re and less lower I^*. The object oscillates from side to side with a well defined period.

Figure 2.5: Measured trajectories of primitive motions from experiments, from left to right: steady decent, periodic tumbling, transitional chaotic, periodic fluttering observed by [Field et al., 1997], and transitional helix, periodic spiral motions observed by [Zhong et al., 2011]. The color in the last two figures show the distance of the trajectories in vertical direction.

- **Transitional Helix motion (TH)** The object falls in a helical path at a constant speed. This transitional motion is found between the fluttering and spiral regimes in the phase diagram (Figure 2.4 (e)).
- **Periodic Spiral motion (PS)** The object falls circularly in three dimensional space at higher Re and lower I^*.

2.5 Numerical simulation in quasi-2D setup

2.5.1 Simple phenomenological model

In the paper [Tanabe and Kaneko, 1994], a simple phenomenological model is proposed based on the force model including lift and drag forces in two dimensional case. Four primitive motion patterns are found with different drag coefficients.

In X-Y plane (Figure 2.6), (x, y), θ are the position and angle of the center of mass of a falling object. (u, v) and ω are the linear and angular velocities of the object, obviously, $u = \dot{x}, v = \dot{y}, \omega = \dot{\theta}$. Let m be the mass of the object, calculated by the density and aspect parameters, for example, a rectangular paper $m = \rho_s Lab$. The $A_\perp$ and $A_\parallel$ are the drag coefficients in perpendicular and parallel directions to the falling object.

The drag forces at the center of mass in the perpendicular and parallel directions of

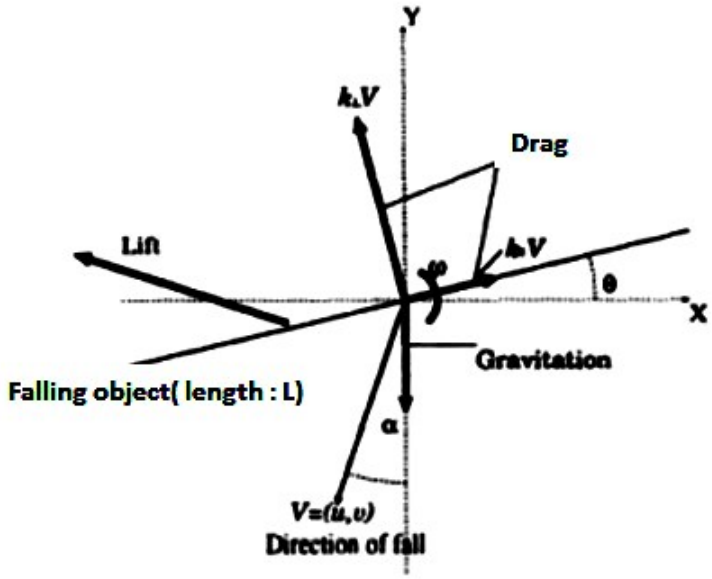


Figure 2.6: The force model including lift and drag forces in 2D, modified figure from [Tanabe and Kaneko, 1994]

the object are calculated as follows:

$$\begin{aligned} F_{\perp} &= -mA_{\perp}(vcos\theta - usin\theta) \\ F_{\parallel} &= -mA_{\parallel}(vsin\theta + ucos\theta) \end{aligned} \tag{2.21}$$

In X-Y coordinate,

$$\begin{aligned} F_x &= -F_{\perp}sin\theta + F_{\parallel}cos\theta \\ F_y &= F_{\perp}cos\theta + F_{\parallel}sin\theta \\ F_{\theta} &= -mA_{\perp}\omega L/12 \end{aligned} \tag{2.22}$$

According to Equation 2.8 in the Section Kutta-Joukowski's theorem, the lift force is given by

$$\begin{aligned} L_x &= kL\rho_f \pi V^2 cos\beta cos\alpha \\ L_y &= -kL\rho_f \pi V^2 cos\beta sin\alpha \\ L_{\theta} &= -L^2 \pi V^2 cos\beta sin\beta/4 \end{aligned} \tag{2.23}$$

where $V = \sqrt{u^2 + v^2}, \alpha = arctan(u/v), \beta = \alpha + \theta$ and

if $sign(v)sin\beta \geq 0, k = 1$
if $sign(v)sin\beta < 0, k = -1$

Combining the Equation 2.21, Equation 2.22 and Equation 2.23 by the Newton's second law, the ODEs governing the motion of falling object are achieved:

$$\begin{aligned}
\dot{u} &= -(A_\perp sin^2\theta + A_\parallel cos^2\theta)u + (A_\perp - A_\parallel)sin\theta cos\theta v - kL\pi\rho_f V^2 cos\beta cos\alpha/m \\
\dot{v} &= -(A_\perp cos^2\theta + A_\parallel sin^2\theta)v + (A_\perp - A_\parallel)sin\theta cos\theta u + kL\pi\rho_f V^2 cos\beta sin\alpha/m \\
\dot{\omega} &= -A_\perp\omega - 3\pi\rho_f V^2 cos\beta sin\beta
\end{aligned} \tag{2.24}$$

Four primitive motions are found with the change of parameters $A_\perp$, $f(=\frac{A_\perp}{A_\parallel})$, ρ, L, corresponding to the regimes in Figure 2.4 in two dimension (Figure 2.7). From this figure,

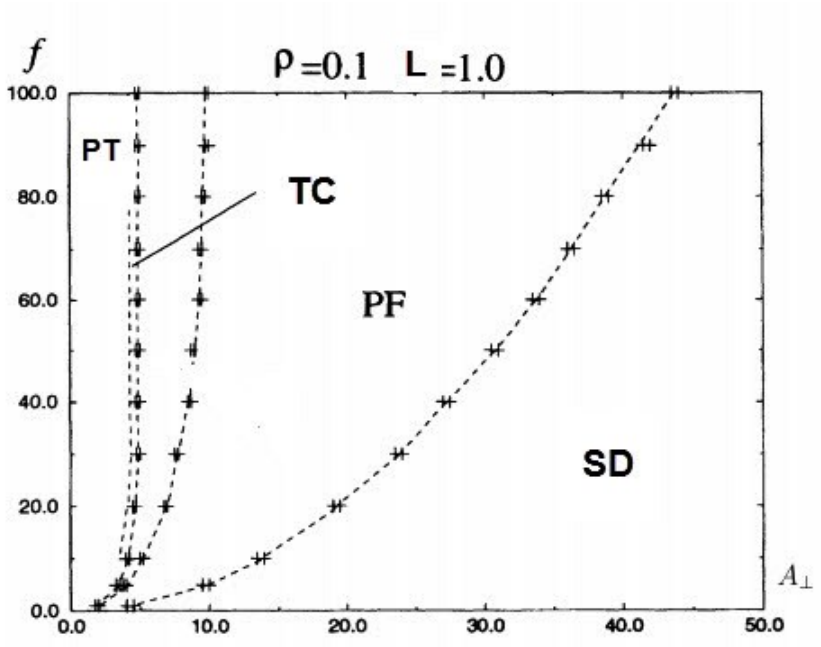


Figure 2.7: 2D phase diagram at $\rho = 0.1$ and $L = 1.0m$. For larger f the regions can clearly be distinguished. As f gets smaller, the chaotic region becomes narrower and finally disappear at $f \approx 1.0$. Modified from [Tanabe and Kaneko, 1994]

we know that different primitive motion trajectories can be approximately calculated from Equation 2.24, so that we will explain how to synthesis primitive motion trajectories in Chapter 3 by solving the ODEs.

2.5.2 Phenomenological model extension

Belmonte et al. [Belmonte et al., 1998] present a simple experiment on flat strips dropped in a quasi-2D experimental setup. The strips are set with the same thickness $b = 0.1$-0.2cm, and width $w = a = 0.75$cm. They observed two fundamental motions: side-to-side rotation (tumbling) and end-over-end (fluttering). They defined the Froude number given by the ratio of characteristic times of the downward motion and the rotational flutter.

$$Fr = \sqrt{\frac{m}{\rho_s L^2 w}} \tag{2.25}$$

From experiments, when the angle of a falling object $\theta > \pi/2$, the free fall motion changes from fluttering to tumbling motions while $Fr_c = 0.67 \pm 0.05$ (see Figure 2.8 (c) and (d)). Different with the Tanabe's phenomenological model, the drag force is proved

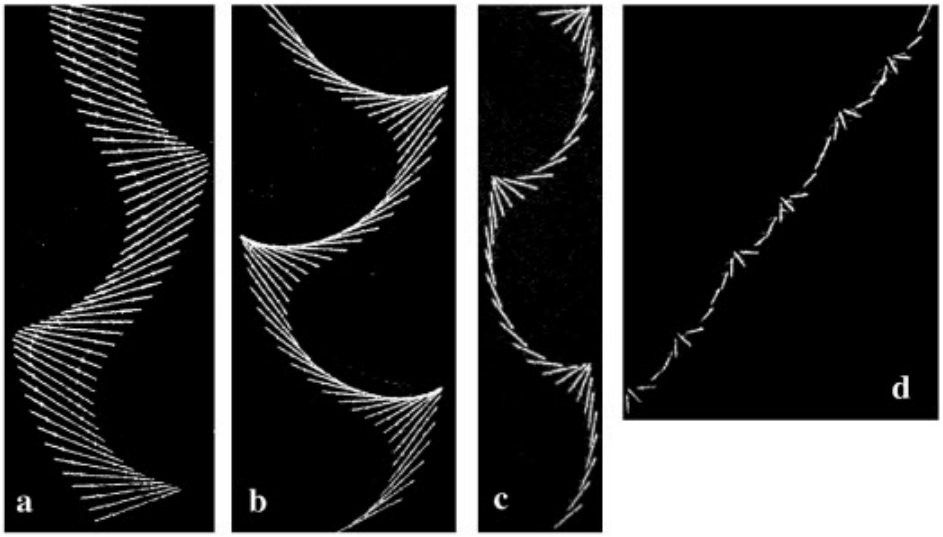


Figure 2.8: collage of consecutive free falling motions in video fields for strips falling in water: (a) L=5.1 cm, m=2.9 g (Fr=0.37); (b) L=4.1 cm, m=2.7 g (Fr=0.45); (c) L=2.0 cm, m=1.4 g (Fr=0.65); (d) L=1.0 cm, m=0.7 g (Fr=0.89) [Belmonte et al., 1998]

to be quadratic in velocity through experiments. Here we use the same notations in the subsection 2.5.1. First the force equations in the perpendicular and parallel direction to the falling object are redefined.

$$\begin{aligned} F_\perp &= -A_\perp \rho w L V^2 cos(\alpha - \theta) \\ F_\parallel &= -A_\parallel \rho w L V^2 sin(\alpha - \theta) \\ F_\theta &= -A_\theta \rho w L V^4 \omega^2 \end{aligned} \tag{2.26}$$

where $A_\theta = 0.0674$, so the ODEs (Equation 2.24) is modifies to the form as follows.

$$\begin{aligned} \dot{u} &= (A_\perp sin\gamma sin\theta - A_\parallel cos\gamma cos\theta + 4\pi |sin\gamma| cos(\alpha \pm \frac{\pi}{2})) \frac{V^2}{Fr} \\ \dot{v} &= -(A_\perp sin\gamma cos\theta + A_\parallel cos\gamma sin\theta - 4\pi |sin\gamma| sin(\alpha \pm \frac{\pi}{2})) \frac{V^2}{Fr} - \frac{1}{Fr} \\ \dot{\omega} &= \pm A_\theta \frac{12\omega^2}{Fr^2} - 6\pi V^2 sin(2\gamma) \end{aligned} \tag{2.27}$$

where $\gamma = \alpha - \theta$. It is proven that the transition from fluttering to tumbling is decided by the Froude Number Fr. This extensional model reproduce the transition, and present the Fr similarity of the free fall motions more clearly.

Chapter 3

Motion Modeling

Motion modeling is an essential step for free fall motion synthesis. In order to synthesize the global path of a falling object, we need to obtain primitive motions which are the feature characteristics of free fall. Data-driven method is adopted in this work as pre-computed database technique. These motion synthesis approaches are often used in character animation, nevertheless, this research firstly considers complex natural phenomena as autonomous motion by capturing physical features.

3.1 Introduction

In this chapter, we discuss the simulation steps in building pre-computed trajectory database for primitive motion synthesises. This issue in motion modelling are threefold: motion segmentation and clustering, motion database and primitive motion synthesis, each of which is discussed in the following section respectively.

Notice that here talk about the motion modelling of six primitive motions of free fall motions, which are described in the previous chapter, i.e. steady descent, tumbling, chaotic, fluttering, helix and spiral motions. We classify similar motions into motion groups labelled with different group specification. The motion classification and motion graph about motion modelling will be interpreted in Chapter 4.

3.2 Motion segmentation and clustering

3.2.1 Motion segmentation

It is not easy to build a trajectory database of free fall motion because capturing the accurate trajectories of light-weight objects from reality seems to be infeasible, the main reasons are as follows.

- The free fall motion used to be fast (release form 1.5 meter height, about 1 second) and wide-ranged with long distance. Even using the high-speed cameras, multiple cameras need to be set and the analysis work is tedious and labour-consuming .

- Also, the motion sometimes goes to chaotic motion, the captured data will not be available for common use and the results are inaccurate.
- It is impossible to use 3D markers in acquisition system due to the light-weight attribute of falling object.

Here, we apply three approaches to obtain motion segments and choose the best alternative by comparing with each other.

Fluid simulation

To solve the Navier-Stokes equations, stable fluid with Semi-Lagrange [Stam, 1999] and Smoothed Particle Hydrodynamics (SPH) [Müller et al., 2003] are fine alternatives. Considering the vortex field in the flow, the vortex particle is used to represent the swirling area on the basic flow[Park and Kim, 2005]. The trajectory of a particle starts from tracing the vortex particle from frame to frame by following velocity vectors until last frame or the boundary has been reached.

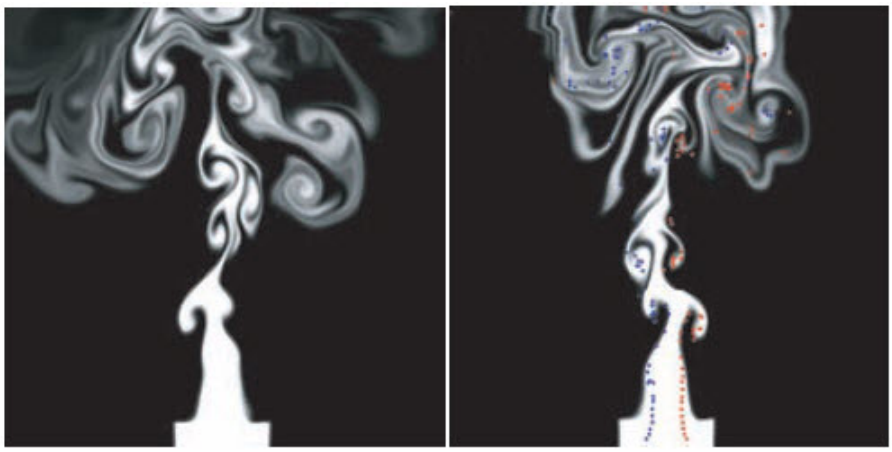


Figure 3.1: Rising smoke simulation by using vortex particles, which are the colourized dots in the right figure. [Park and Kim, 2005]

Although it is a common way to use the similar approaches in most recent works for falling simulation in gaseous fluid([Shi et al., 2005], [Vázquez and Balsa, 2008]), we do not adopt this approach, because

1. It cannot capture the nature of free fall motions, not be able to detect all the primitive motions in our phase diagram.
2. The angular velocity of object cannot be achieved correctly.
3. Required motion trajectories are difficult to be satisfactorily captured.

Phenomenological model

In Chapter 2, we explained the ODEs Equation 2.24 based on Kutta-Joukowski theorem. The Equation 2.24 is modified into the following clear form.

$$\begin{aligned}
\ddot{x} &= -(A_\perp sin^2\theta + A_\parallel cos^2\theta)\dot{x} + (A_\perp - A_\parallel)sin\theta cos\theta\dot{y} - kL\pi\rho_f V^2 cos\beta cos\alpha/m \\
\ddot{y} &= -(A_\perp cos^2\theta + A_\parallel sin^2\theta)\dot{y} + (A_\perp - A_\parallel)sin\theta cos\theta\dot{x} + kL\pi\rho_f V^2 cos\beta sin\alpha/m \\
\ddot{\theta} &= -A_\perp\dot{\theta} - 3\pi\rho_f V^2 cos\beta sin\beta
\end{aligned} \tag{3.1}$$

where $V^2 = \dot{x}^2 + \dot{y}^2$, (x, y, θ) are the position and orientation of a falling object. We employ the standard fourth-order Runge-Kutta algorithm to solve the second-order ordinary differential equations Equation 3.1 using the variables $x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}$. The initial condition is set to be $x_0 = x_r, y_0 = y_r, \theta_0 = \theta_r$. (x_r, y_r, θ_r) are the released position and the initial angle of the object. The solved result of fluttering motion is shown in Figure 3.2.

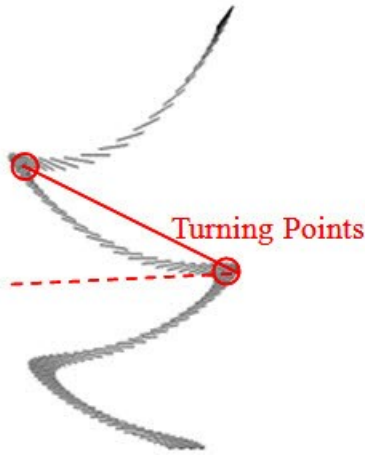


Figure 3.2: The fluttering trajectory solved by ODEs, $A_\perp = 4.1, A_\parallel = 0.9$

The turning points (Figure 3.2) are used to segment the result curves. Because at turning points, the distance in x direction comes to the maximum and the angle turns to opposite orientation, so the following constraints are required:

$$\dot{x} = 0, \dot{\theta} = 0 \tag{3.2}$$

Harmonic functions

Because of the symmetry of the fluttering motion, the frequency of oscillatory component of vertical velocity is twice of the horizontal velocity component. According to the measured data of experiments, we use harmonic functions to describe the generally complex planar motions.

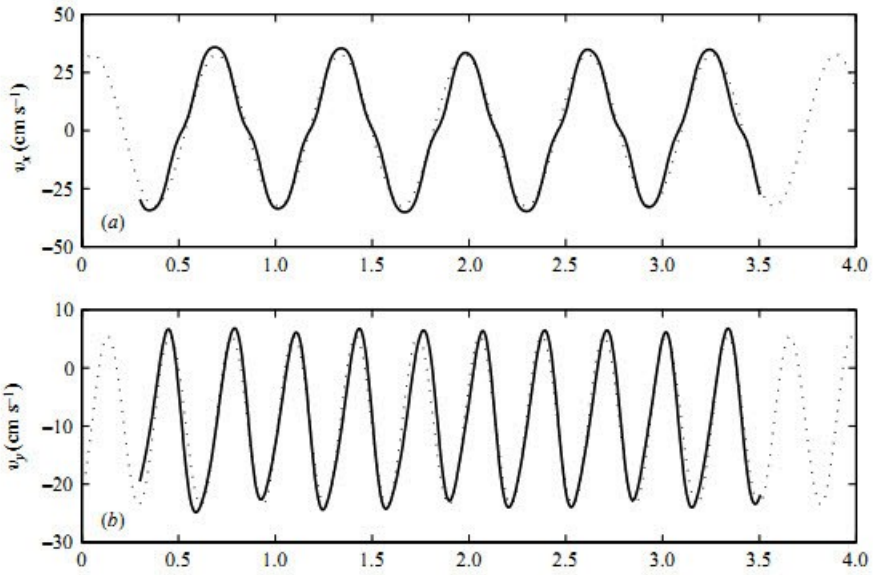


Figure 3.3: Measured Data by [Andersen et al., 2005], $\dot{x}$ velocity in x direction(top); $\dot{y}$ velocity in y direction

The harmonic functions are defined as follows:

$$\begin{aligned} x_t &= x_0 - \frac{A_x}{\Omega} sin(\Omega t) \\ y_t &= y_0 - Ut - \frac{A_y}{2\Omega} cos(2\Omega t) \end{aligned} \tag{3.3}$$

where A_x and A_y are the amplitudes of vertical and horizontal velocities caused by the oscillation of falling object due to the viscous flow around. Ω describes the angular frequency of the motion. U is calculated by Equation 2.15 in Chapter 2. From Equation 3.2, the turning points of the trajectory are given at time-steps $t_i = \frac{2k+1}{2\Omega}\pi, k \in Z \geq 0$.

Comparing the phenomenological model and harmonic functions approaches, both of these approaches are segmented by the turning points (Figure 3.4), by tuning parameters $A_\perp$ and $A_\parallel$ in Equation 3.1; A_x, A_y, U, Ω in the Equation 3.3.

The controllability of phenomenological model is weaker than harmonic functions because it is difficult to get a qualified trajectory from phenomenological model (Figure 3.5). The required trajectory segments are queried possibly and quickly by harmonic functions. The disadvantage of harmonics function is no orientation info by the equations. In the next step, we interpolate the angle of object by combining these two approaches.

3.2.2 Segments clustering

After the step of segmentation, there are numerous of segments obtained by changing parameters A_x, A_y, U, Ω in Equation 3.3 in a designated step (Figure 3.6). We use K-means

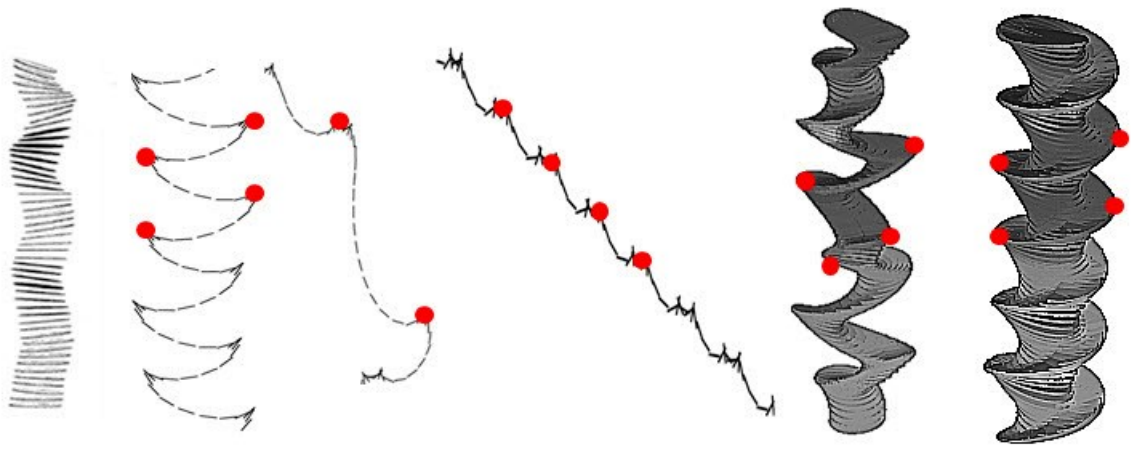


Figure 3.4: Six primitive motion trajectories, from left to right: steady descent, fluttering, chaotic, tumbling, helix and spiral motion(measured data from [Andersen et al., 2005] and [Zhong et al., 2011]). The red points are distinguished as turning points.

clustering algorithm by open software KMlocal, Version: 1.7.3[Mount, 2005] for clustering segment data using feature vectors.

The input segments set $(S_i|i = 1, 2...N)$,N is the number of segments, are classified based on the value of feature vectors of each segment from the start feature point P_i^0 to the end feature point P_i^1. Feature vectors set $V\{V_i = P_i^1 - P_i^0, i = 1, 2...N\}$ (Figure 3.5) are assigned into K classes using K-means algorithm.

3.3 Pre-computed trajectory database

After segments clustering process, we store positions p_i^k and orientations q_i^k of the i-th feature points in k-th segments in segments set S into trajectory database (Figure 3.7). $\{(p_i^k, q_i^k)|i = 1, 2...M\}$, M is the number of frames of the segment S_k, is the point info per segment.

About the orientation of falling object, we use the calculated results of phenomenological model for interpolation. Let $\{(p_i^1, q_i^1)|i = 1, 2...Q\}$ is the calculated segment after clustering with Q frames and the feature Vector V^*. We calculate the feature Vector V_k of segment S_k.

If the angle of V_k is less than the angle of V^*, i.e, $arctan(\frac{p_k y^1 - p_k y^0}{p_k x^1 - p_k x^0}) \leq arctan(\frac{p_y^1 * - p_y^0 *}{p_x^1 * - p_x^0 *})$, then we find point j $(1 \ll j < Q)$, let $V_j \leq V_k < V_{j+1}$.

We interpolate q^k with $\{q_i^1|i = 1, 2...j\}$ by comparing positions $\{p_i^k|i = 1, 2...M\}$ and $\{p_i^1|i = 1, 2...j\}$ using linear interpolation.

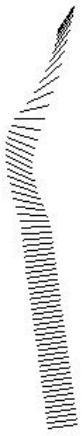


Figure 3.5: The trajectory by solving Equation 3.1 , $A_{\perp} = 4.6, A_{\parallel} = 0.15$

If the angle of $V_k > V^*$, we divide the segment S_k into smaller parts and let the angle feature vector of every parts is not larger than the angle of V^*, then use the same process of linear interpolation.

3.4 Motion synthesis of primitive motions

In this section, we give the details how to synthesis the primitive motions of free fall (Figure 3.4): steady descent, tumbling, fluttering, chaotic, helix and spiral motions, which are incorporated into making the initial trajectories for given input about the object and fluid.

According to the measured data in Figure 3.4, we utilize a set of additional qualifiers for each word to characterize a primitive motion, as shown in Table 3.1. There are slide left, slide right in two dimensional plane, and turn left, turn right, front, back in three dimensional space. Notice that the difference between the helix and spiral motions is the curvature of the trajectory, even though they have the same motion qualifiers.

Table 3.1: motion repertoire table

motion aspects	motion qualifiers
steady descent	downward
fluttering	glide left & glide right
chaotic	glide left & glide right | glide left
tumbling	(glide left & glide left)|(glide right & glide right)
helix	turn left & turn right, front, back
spiral	turn left & turn right, front, back

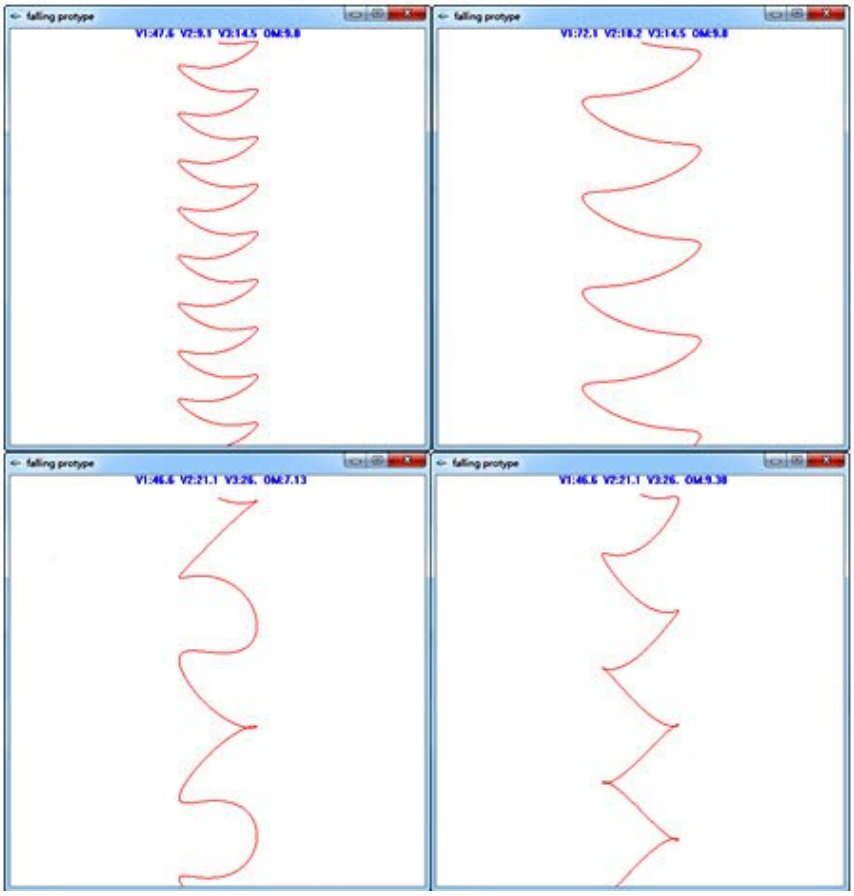


Figure 3.6: Four trajectories calculated by Equation 3.3, top-left: $A_x = 47.6, A_y = 14.5, U = 9.1, \Omega = 9.8$; top-right: $A_x = 72.1, A_y = 14.5, U = 18.2, \Omega = 9.8$; bottom-left: $A_x = 46.6, A_y = 26.0, U = 21.2, \Omega = 7.13$; bottom-right: $A_x = 46.6, A_y = 26.0, U = 21.1, \Omega = 9.38$.

3.4.1 Trajectory search tree

In this subsection, the motion trajectories of fluttering, tumbling and chaotic motions are synthesized by a tree structure which conforms to the experimental measured data. We compare the trajectories of chaotic motion, periodic fluttering and tumbling motions in experiments. Because the airflow behind the free fall object reveals vortex shedding, turbulence and other complex motions, the object comes to turning points, where the angular velocity becomes 0 and the velocity in the oscillation direction also be 0 but in vertical direction be a maximum value. The object faces to two alternatives of sliding left or sliding right (fluttering or tumbling motion).

In Figure 3.8, we create a trajectory search tree, where start point is set as root and turning points as nodes. The edge between nodes is the motion segments from motion sequences. Note that the chaotic motion is defined by using different kinds of nodes, in

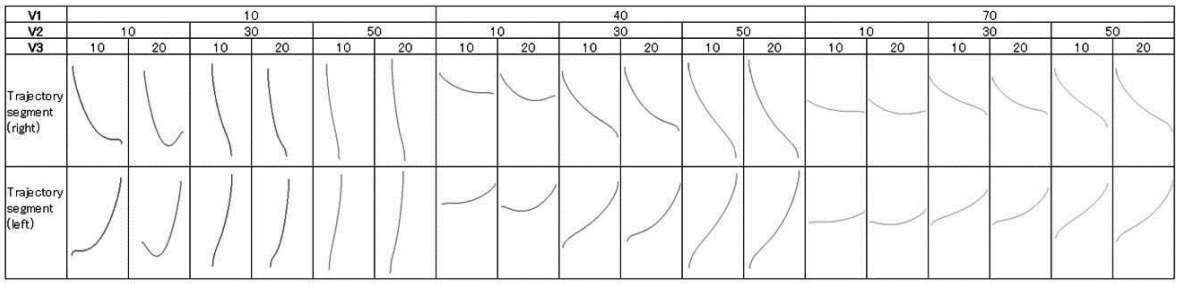


Figure 3.7: part of pre-computed trajectory database

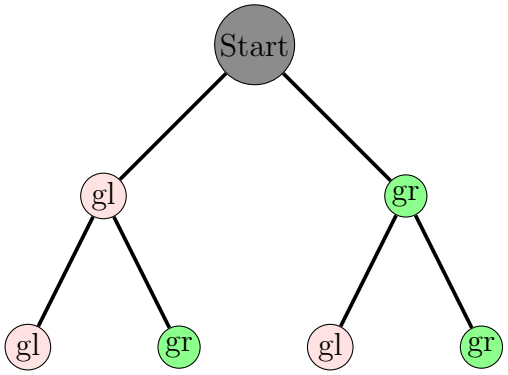


Figure 3.8: The first two levels of a small trajectory search tree.

other words, chaotic motion is the transition region between periodic fluttering and periodic tumbling motion in phase diagram. It is noticed that chaotic motion happens when the angle at turning points comes to orient vertically, the dynamic is much more sensitive to experimental environment (including flow movement) than in the periodic regions (PF & PT in phase diagram). A sample structure of fluttering, chaotic, tumbling motion is shown in Figure 3.9.

3.4.2 Unified harmonic functions

When projecting the motion paths into XY plane which are measured by [Zhong et al., 2011], we notice that the curves of six primitive motions have characteristic shapes: steady descent motion trajectory is likely to be one point; fluttering, tumbling and chaotic motion trajectories are in a straight line; Spiral motion trajectory is similar to circle and helix motion trajectory is similar to eight petals rose curve, as shown in Figure 3.10. In the XY plane, free fall motion is consisted with two parts of motions:

1. Rotation about the vertical falling direction

2. A harmonic elliptical oscillation which oscillates in both x and y direction simultaneously.

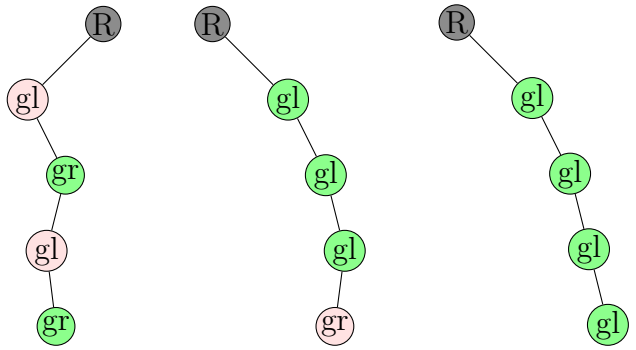


Figure 3.9: The tree structures of fluttering, chaotic, tumbling created by traversal of four levels search tree, which are corresponding to the trajectories in Figure 3.4

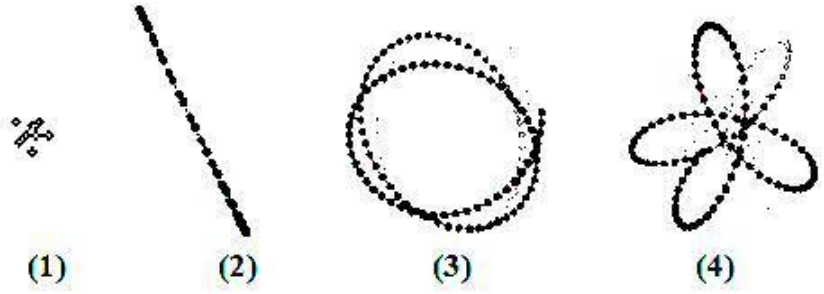


Figure 3.10: Measured curves projected to the XY plane, (1) steady descent; (2) fluttering& chaotic& tumbling; (3) spiral; (4) helix

All the curves are expressed as following equation:

$$\begin{aligned} x_t &= A_e cos(\Omega t)(1 + \epsilon_e sin(k\Omega t)) \\ y_t &= A_e sin(\Omega t)(1 + \epsilon_e sin(k\Omega t)) \\ z_t &= h - Ut \end{aligned} \tag{3.4}$$

where A_e is the amplitude of elliptical oscillation generated in XY plane orientation. ϵ_e is the aspect ratio of the short axis and long axis of oscillation ellipse. k is the ratio of the elliptical oscillation to the period of rotation of the falling object. Ω is the angular frequency of the falling motion.

Deduced from the Equation 3.4, a simple form of free fall trajectory is as follows.

- $\epsilon_e \to 0, k = 1 \mapsto$Spiral motion
- $A_e \to 0, k \to 0 \mapsto$Steady descent motion
- $\epsilon_e \neq 0, k = 4 \mapsto$Helix motion

corresponding to the Figure 3.10.

3.4.3 Implementation result

In Figure 3.11, we compare the synthesized trajectory with the measured data, and prove that the interpolated angles from Equation 3.1 make the simulation result more realistic.

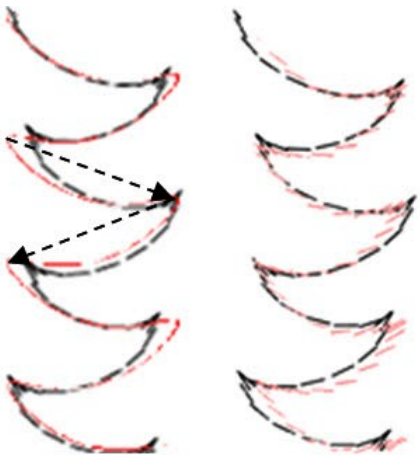


Figure 3.11: Comparison synthesized trajectory (Red) with measured data (Black) of fluttering motion, and the result of no orientation (left) and interpolated orientation (right). The arrow lines are feature vectors.

Because of the nature of chaotic motion, we synthesize the motion randomly with feature vector $V = rV_0$ and amplitude of oscillation $A_e = rA_0$. r is a random number between 1/10 and 10 calculated by Box-Muller method, the angle of V_0 is the initial feature vector researched by the release angle of the object from trajectory database, and A_0 is the initial amplitude of object oscillation given in the Chapter 4.

After building pre-computed trajectory database, six primitive motions are possible to be synthesized using trajectory search tree and unified harmonic functions (Figure 3.12). Feature vectors (Figure 3.11) are the main key for searching trajectory database to obtain the motion segments.

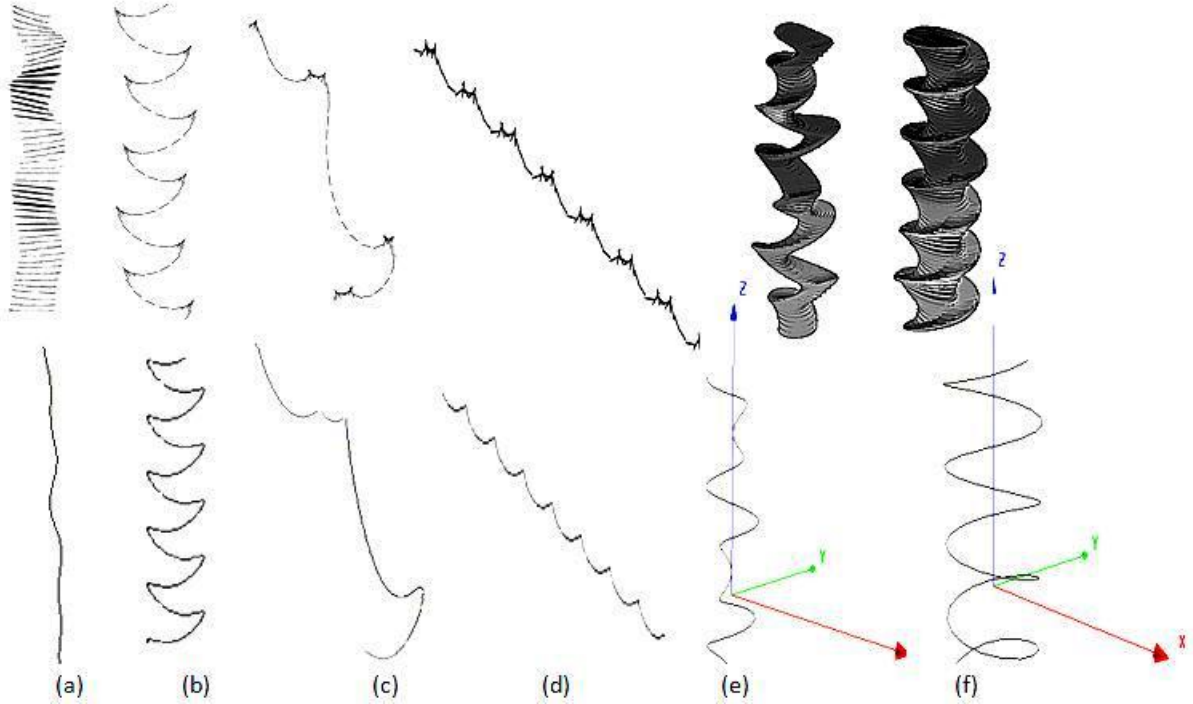


Figure 3.12: Six measured trajectories (top, worked by [Field et al., 1997] [Andersen et al., 2005] [Zhong et al., 2011]) and synthesized trajectories by our work (bottom) of the primitive motions in free fall which are described in the phase diagram, (a) steady decent, (b) fluttering, (c) chaotic, (d) tumbling, (e) helix and (f) spiral motions

Chapter 4

Motion Synthesis

In this chapter, our focus is on the free fall motion in three-dimensional environment. We try to find out the rule how to connect primitive motions together to represent the realistic motion paths. First we introduce the specific motion classification for free falling motions. Given the Reynolds number and the dimensionless moment inertia, we indicate the possible permutations of primitive motions. After then, the free fall graph is constructed for combining the motions. Finally, we give some details of implementation to obtain a realistic and natural simulation. Most of our analysis work here is based on the experimental data of [Razavi, 2010], which carries out more than 6000 experiments of falling leaves.

4.1 Motion classification

In Chapter 3, the primitive motions $\{L_i|S_1, S_2, ...S_k\}$ (S_k is k-th segment in primitive motion L_i) are synthesized by motion segments S_n from pre-computed trajectory database. Because primitive motions are the basic motions in free fall through various experimental works, the motion groups $\{G_i|1 \leq i \leq 6\}$ are used to represent the basic and similar motions of free fall motion.

For a free fall motion M, $M\{M = m_1 \parallel m_2 \parallel ...m_i\}$ is annotated with a label, which is set to be L_i, where L_i is primitive motion corresponding to the motion repertoire table Table 1 in Chapter 3. We define that, L_1: steady descent motion(SD);L_2: fluttering motion(PF); L_3: chaotic motion(TC); L_4: tumbling motion(PT); L_5: helix motion(TH); L_6: spiral motion(PS), as shown in Figure 4.1.

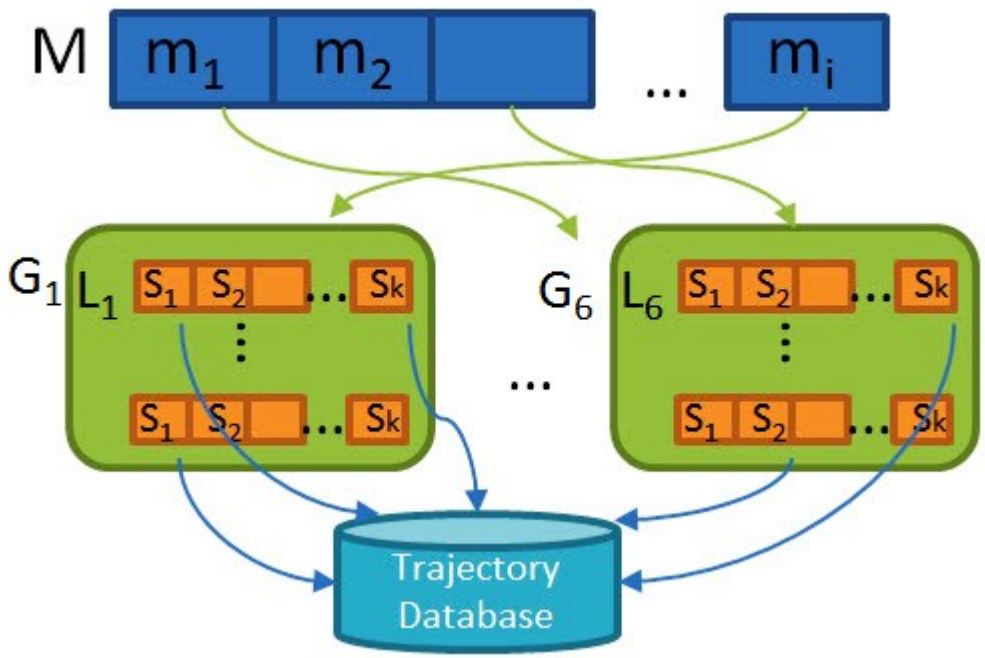


Figure 4.1: Motion classification, blue: motion classes; green: motion groups; Brown: motion segments

Based on thousands of experiments that 300 elliptical papers are released from 125cm height to the ground, the free fall trajectories are classified into 7 motion classes (Table 4.1).

Table 4.1: Motion classes table

Class 1	L_4
Class 2	$L_3 \rightarrow L_6, L_3, L_4 \rightarrow L_6$
Class 3	$L_1 \rightarrow L_2, L_1 \rightarrow L_6$
Class 4	$L_4 \rightarrow L_6, L_3$
Class 5	$L_2 \rightarrow L_6$
Class 6	L_4, L_6
Class 7	L_1

Because the classification is only based on the primitive motions $\{L_1, L_2, L_4, L_6\}$, the two transitional motions, i.e. chaotic and helix primitive motions, are not clearly distinguished in the experiments of [Razavi, 2010]. The reasons could be speculated as that the chaotic motion looks like permutation of fluttering and tumbling motions and helix motions are similar to spiral motions. The difference between the transitional motions and other motion is also extremely apparent due to no periodicity in transitional motions. In spite of the uncompleted taxonomy, the experimental results are valid for our basic analysis.

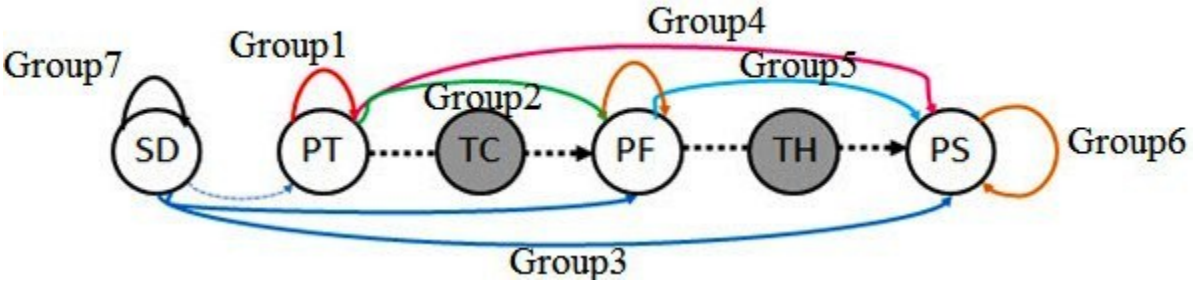


Figure 4.2: Motion classes in linked nodes including transitional motions

In Figure 4.2, seven motion groups are recorded which are illustrated in different colors. The arrow of the lines means the motion transition between different primitive motions. If the arrow pointed itself, it means the motion happened independently. For example, the green link means that the object first tumble and then flutter in a order; the black link means that only steady descent happened. In these experiments, Re is about 102, and $I*$ is 10^{-3}, so the main motion is PS by looking up the phase diagram. From the motion classification from experimental data, we conclude a hypothesis as follows.

Hypothesis 4.1 *If* $M\{M = m_1 \parallel m_2 \parallel ... \parallel m_i\}$ *is a free fall motion in 3D, then* $m_i \in \{L_j | 1 \leq j \leq 6\}$ *and the subscript sequence* $\{j_1, j_2, ..j_i\}$ *should be an increasing sequence.*

To prove this hypothesis, we analysis it qualitatively. When an object starts falling from released point, the vortexes are generated gradually behind the object due to the vorticity of around flow. The free fall motion becomes more and more sensitive to the internal forces, including drag and lift forces. According to the Subsection 2.4.3 ($Re - I^*$ Phase diagram) in Chapter 2, the sensitivity obeys the change while increasing Re and decreasing I^* . So we consider that it is impossible changing from spiral motion into steady descent motion. Unfortunately, the quantitative explanation of this hypothesis is still unsolved [Zhong et al., 2011].

Another way to verify the hypothesis, the free fall motion can be obviously observed by experiments. From thousands of experiments in the work [Razavi, 2010], the hypothesis fits the experimental data perfectly. In our experiments of the Chapter 5, we also prove the validity of the hypothesis.

From the hypothesis, the number of combinations of all primitive motions is revealed by,

$$N = \sum_{i=1}^{i \leq k} C_k^i, k \in Z, k \in [1, 6] \tag{4.1}$$

where k is the level of the main primitive motion by looking up phase diagram.

4.2 Markov chain model

In Section 4.1, we make it clear that the free fall motion trajectory is the connection of primitive motions from motion groups. We focus on how to decide the m_i for the motion M. The first-order discrete-time Markov chain model is proposed for solving this issue.

Markov chain is a class of stochastic processes that are distinguished by Markov property and often applied in biology, engineering and economics [Kijima, 1997]. Let consider a discrete-time stochastic process $\{X_n\}$ with $N_0 \in Z \equiv i \in [1, 6]$ as the state space, which is corresponding to motion groups $\{G_i | 1 \leq i \leq 6\}$ (Figure 4.1). Markov property asserts that the distribution of random variable X_{n+1} in process $\{X_n\}$ depends only the current state $X_n = i_n$, other than the whole history $\{X_0 = i_0, ...X_n = i_n\}$.

$$P[X_{n+1} = j | X_n = i_n] = P[X_{n+1} = j | X_0 = i_0, ...X_n = i_n] \tag{4.2}$$

where $j, i_0, ..., i_n \in N_0$. The stochastic process $\{X_n\}$ is called a Markov chain. Let the

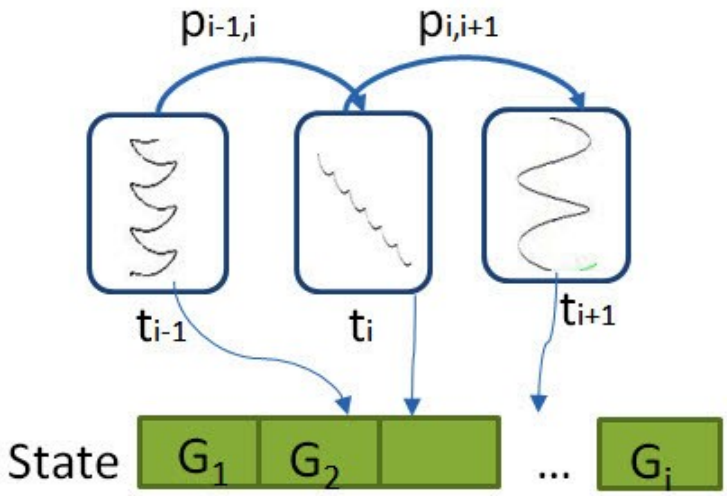


Figure 4.3: Markov chains model

state space N_0 be the motion groups G and process $\{X_n\}$ on $\{X_t\}$ discrete time set, the transition probability $p_{ij} = P[X_{t+1} = L_j | X_t = L_i]$ is the conditional probability from primitive motion L_i to primitive motion L_j. The transition matrix is given as $P = (p_{ij})$ (Figure 4.3). Because the process $\{X_t\}$ is stochastic, the matrix requires:

$$p_{ij} \geq 0, and \quad \sum_j p_{ij} = 1 \quad i, j \in [1, 6]$$

Also according to the hypothesis in Section 4.1, P has the following form.

$$P = \begin{pmatrix} Q & R \\ 0 & T \end{pmatrix}_{i \times j} \tag{4.3}$$

Then, we discuss about the realization of Markov chain $\{X_t\}$ and transition matrix P calculation.

In order to obtain the process $\{X_t\}$, the model starts with initial state at time $t_0 = 0$. Then the iteration step is executed, for the state L_i at time t to state L_j at $t + 1$, the calculation depends on the probabilities at the i-th row of transition matrix P, i.e, $P_i = (p_{ij}|j = 1, 2..., 6)$.

The state transition probabilities for the transition matrix P are found by counting the state transitions happened in the experimental data. Let set N_i is the number of all transitions from state L_i in the experimental data, N_{ij} is the number of transitions state L_i to state L_j, the probability is given as $p_{ij} = \frac{N_{ij}}{N_i}$.

The advantages of using to first-order discrete time Markov chain model is as follows.

- Comparing to the high order Markov chain model and dynamic Bayesian network, for example, second order Markov model $P[X_{n+1} = j|X_{n-1} = i_{n-1}, X_n = i_n$, the transitions betweens primitive motions $\{X_t, t \in T\}$ do not have common parameters for different primitive motions due to the high-level representation of motions. Also, the next motion is only related to the current state.

- The state space is based on the primitive motions rather than segments or points in a trajectory, the characteristic features of each primitive motion are apparent, so that it is effortless to obtain the valid transition matrix from experimental data for realistic simulation of free falling motions.

- The computation cost for the model is light for real-time simulation as on-line operation.

4.3 Graph construction

The special free fall motion graph (Figure 4.4) is based on the approach of motion graph in the work [Kovar et al., 2002]. This structure is a completed directed graph, in other words, each node of a object graph is connected to other nodes in the same graph. We use $G = (V, E)$ to represent this structure, V is the node set; E is the edge set.

Every frame in these motion sequences of primitive motions can be seemed as the node in the motion graph; the edge between nodes is a transition splice in sequences. We search the graph only in one-way from top to down as the order according to the hypothesis in Section 4.1. Consequently, it is impossible that free fall motions become tumbling after spirally falling as an example.

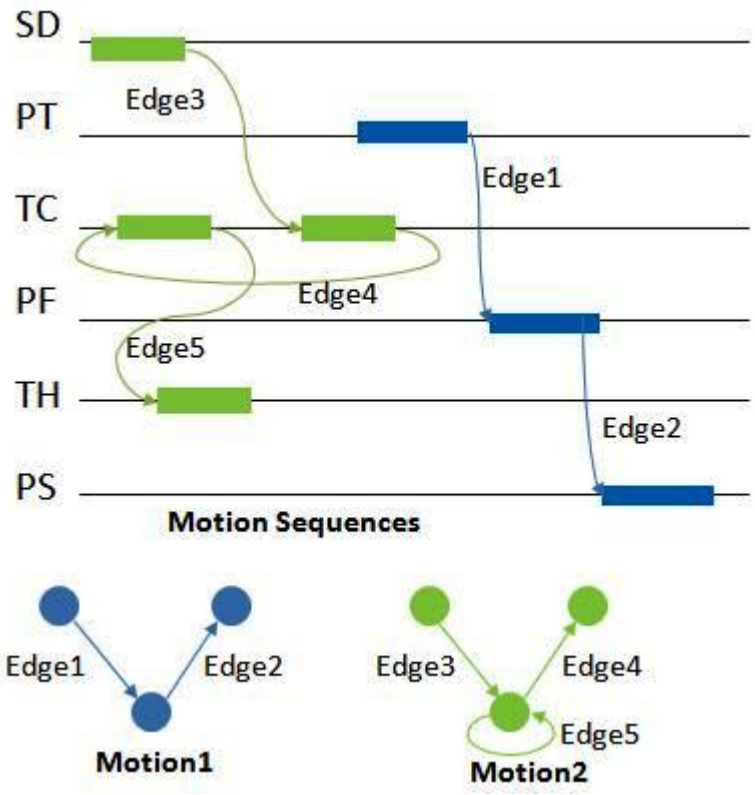


Figure 4.4: Free fall motion graph, a motion path represents a collection of splices between sequences (top) two example motions are represented by the directed graph.

The transition probability is attached to the edge between nodes using the discrete time Markov chain model (Section 4.2).

4.4 Implementation of motion synthesis

In this section, the details of the implementation are described.

4.4.1 Motion specification

From previous steps, we already know the primitive motions used in the motion synthesis, The amplitude and deviation of each motion are demonstrated based on the experimental data from [Razavi, 2010] in this section.

- each motion group is distributed in normal distribution of Gaussian functions (Figure 4.5).

$$f(r) = Ae^{-(\frac{r-B}{C})^2} \tag{4.4}$$

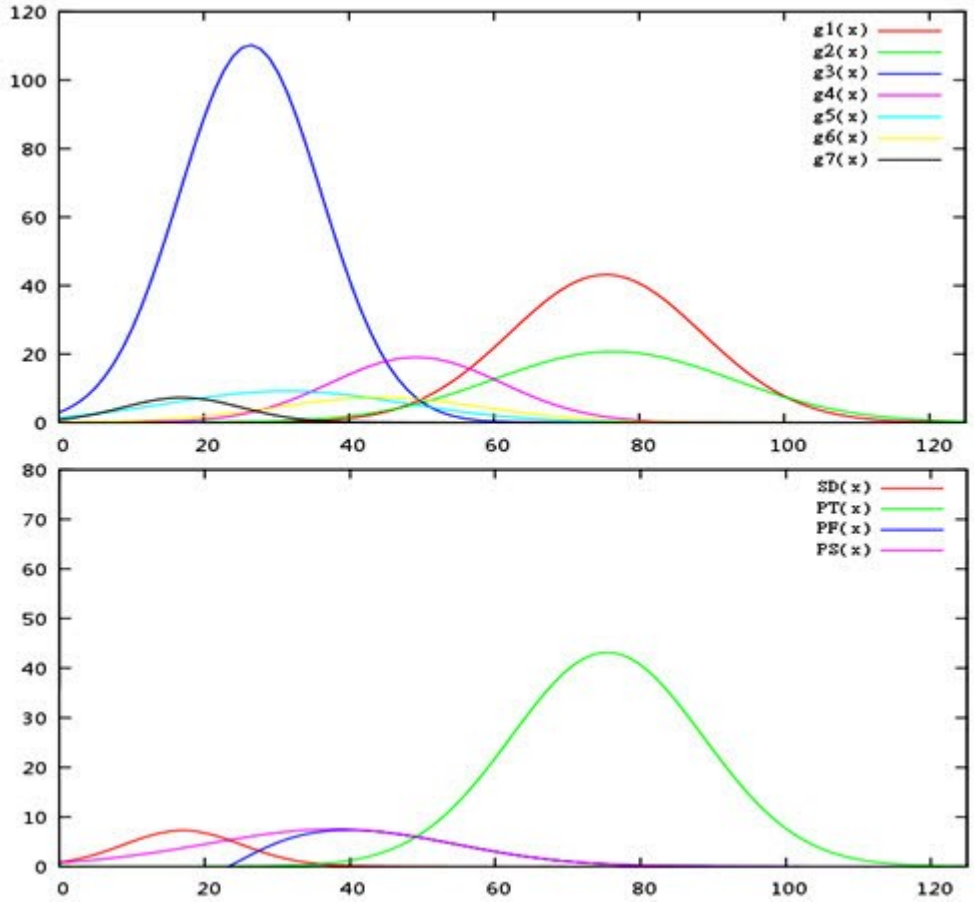


Figure 4.5: Spatial distribution of 7 motion classes (top) ((A, B, C) are (20.7, 76.42, 23.34), (43.16,75.42,18.69), (110.2, 26.41, 13.83), (19.02, 49.27, 16.69), (9.14, 31.36, 23.21), (7.26, 44.22, 20.34), (7.25, 16.99, 11.47) for Class 1∼7) and primitive motions (bottom), horizontal axis is the deviation from released position (cm); vertical axis is the quantity of the object, the released height is 125cm

- If the free fall motions include the tumbling motion, the spatial deviations are linear relationship with the release height h. The reason is that the tumbling motion have the largest deviation in all primitive motions due to its on-side declining motion.

$$d(h) = d_1 h + d_2 \tag{4.5}$$

where $d_1 \sim 1, d_2 \sim 0$.

- Because the spatial deviation $D_i, i \in [1, 6]$ is independent to the size of the shape of a falling object, the aspect ratio of width and length $\frac{a}{L}$ is important for deciding the

spatial deviation of falling motion. Assuming the relationship between aspect ratio and deviation is linear, we simply calculate the deviations of fluttering, chaotic, helix and spiral motions.

$$D_i = \frac{kB_iL}{a} \tag{4.6}$$

where B_i is the mean in Gauss distribution of each primitive motion (Equation 4.4), k is the deviation coefficient.

- About the tumbling motion, the frequency of tumbling motion is give as $\Omega \sim \sqrt{b}/a$, and then the initial amplitude of tumbling motion is

$$A_0 = \frac{d(h)U}{h\Omega} \tag{4.7}$$

where d(h)is the result of Equation 4.5, U is the average falling velocity.

By the Equation 4.6 and Equation 4.7, the amplitudes of primitive motions as a function of released height and object aspects are available for the motion synthesis.

4.4.2 Angular Optimization

Interpolation between segments

When we connect two motion sequences in different primitive motions, C0 continuity is kept easily but C1 continuity is also necessary in this case. We make the motion smooth by using spherical linear interpolation for and linear interpolation for positions. We choose the smoothing domain to be $\pm n$ frames around connection point between two segments A and B.

$\{(p_i^A, q_i^A)|i = Q - n + 1, ..., Q\}$,$\{(p_j^B, q_j^B)|j = 1, 2, .., n\}$

where Q is frame number of segment A. The transition frames are defined between A and B. The k-th ($0 \leq k < n$) frame in the transition region are calculated as follows:

$$\begin{aligned} p_k &= \alpha(k)p^A_{Q+k-n+1} + (1 - \alpha(k))p^B_{k+1} \\ q_k &= slerp(q^A_{Q+k-n+1}, q^B_{k+1}, \alpha(k)) \end{aligned} \tag{4.8}$$

where *slerp* is the spherical linear interpolation function. In order to keep C1 continuity between segments, the specific function $\alpha(k)$ need to satisfy the constraints:

$\alpha(0) = 1, \alpha(n) = 0, \dot{\alpha}(0) = \dot{\alpha}(n) = 0$

The following is a cubic polynomial which satisfies these constraints.

$$\alpha(t) = 2t^3 - 3t^2 + 1 \tag{4.9}$$

Orientation modification

It is difficult to obtain the orientation of a falling object in three dimensional case. The modification of the orientations are proposed empirically by looking up the follow diagram (Figure 4.6). There are six rotation patterns:

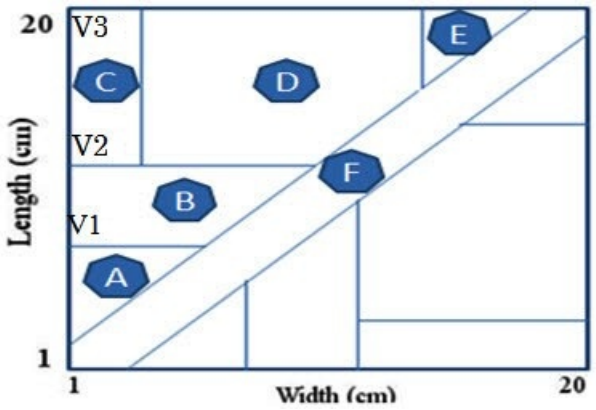


Figure 4.6: The relationship between rotation patterns and width-length ratio

A fall perpendicular

B flutter in width axis → rotation in width axis → rotation in length axis

C flutter in width axis

D flutter in width axis → rotation in length axis

E irregular

F flutter in diagonal axis → rotation in diagonal axis

Where we use rectangle geometry to approximate objects, the width axis and length axis are the lines through center of mass parallel to the direction of width or length. Notice that some rules are concluded from the diagram.

1. when both length and width are small, steady descent motion always happen.
2. most motion starts form fluttering in width axis
3. rotation in length axis is the most stable motion. if the length of falling object is the same with the width axis, rotation in diagonal will become the most stable motion.

The thresholds $V1, V2, V3$ in the figure are defined as: $V1 = V2/2$ means that if the length or width is bigger than $V1$, the object starts oscillating. $V2$ is Froude Number defined in Section 2.5.2. $V3$ is the value of boundary between fluttering regime and chaotic regime in the $Re - I^*$ phase diagram (Section 2.4.3).

Chapter 5

Falling in Wind

The free falling behaviours are sensitive to the wind conditions, because of the dynamics of free fall and the wind flow. Up to this point, the natural and feasible motion synthesis of an object freely falling in quiescent flow is proposed, the influences of wind field to the falling object are interpreted in this chapter. The mathematical modeling of the wind is still a challenging task for the high degree of complexity in computer graphics, it is active research field including many simulations, for example the waving of trees and grasses, and cloth simulations.

To obtain the wind field, the direct and straight approaches are the simulation of the turbulent flow simulation in boundary condition by solving differential equations using Fourier filter [Shinya and Fournier, 1992] [Stam and Eugene, 1993]. The other methods is to simulate the motion in wind field using noise functions (fractional Brownian motion) [Ota et al., 2004] [O.Khorloo and N.Chiba, 2011]. Comparing these flow-based method and noise-based method, the flow-based method provides physically accurate and realistic results but needs high computation cost, the noise-based method is much simpler than the flow-based method and suitable for real time simulation. The disadvantage of noise-based method is losing a physical accuracy. To overcome this problem, the physically-based analysis of wind characteristics is necessary.

This work improves the noise-based approach of [O.Khorloo and N.Chiba, 2011] using the specific spectrum synthesis technique based on inverse Fourier transforms. Our approach expand a two-dimensional wind field to the varying height in terms of logarithmic wind law to achieve an efficient real-time free fall simulation.

5.1 $1/f^{\beta}$ Noise

Fractional Brownian motion (fBm), also known as $1/f^{\beta}$ noise, is observed in various natural phenomena. In computer graphics, fBm is a convenient tool in simulation of fractal images, including mountains, clouds [Peitgen and Saupe, 1988].

The common fBm generation method is the spectral synthesis method (also known as the Fourier filtering method), which is based on the spectral representation of a stochastic process $X(t)$. In a finite time domain $t \in (0, T)$, $X(t)$ is represented as:

$$X(t) = \int_{-\infty}^{\infty} F(f) e^{2\pi i t f} df \tag{5.1}$$

where f is the frequency and $F(f)$ is the Fourier transform of $X(t)$.

$$F(f) = \int_{0}^{T} X(t) e^{-2\pi i t f} dt \tag{5.2}$$

t is measured in seconds, and f is measured by hertz. Equation 5.1 and Equation 5.2 are two different representation of the same Fourier transform, called inverse and forward Fourier transform respectively.

$|F(f)|^2 df$ is the total energy of $X(t)$ in the frequency domain from f to $f + df$, so the average power of X is given by

$$\frac{1}{T} \int_{-\infty}^{\infty} |F(f)|^2 df \tag{5.3}$$

$S(f) = \frac{1}{T}|F(f)|^2$ is the power spectral density function of $X(t)$. When time t goes to infinity, the limit of $S(f)$ is spectral density function.

fBm is represented if the density function has the following form,

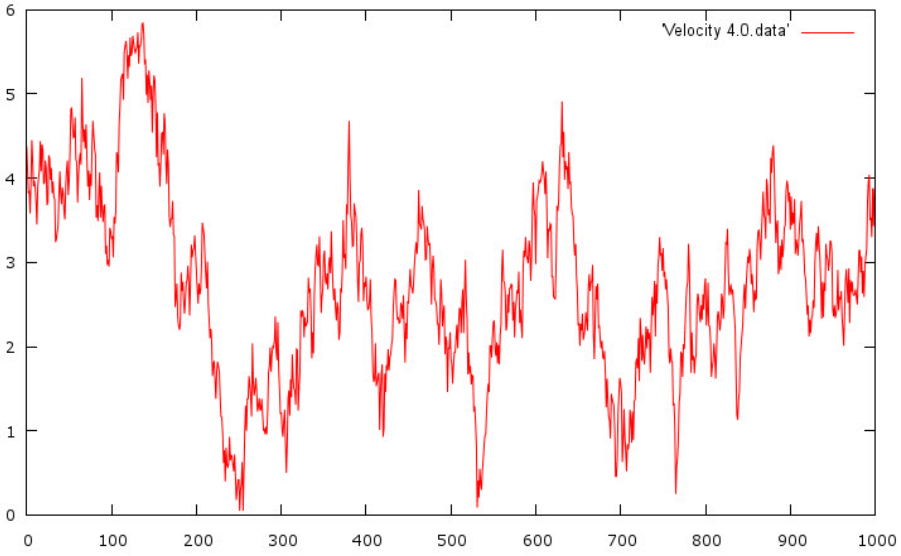
$$S(f) = \frac{1}{f^{\beta}} \tag{5.4}$$

where β is between 1 and 3. The cases of white noise ($\beta = 0$) and random walk process (or Brownian motion, $\beta = 2$) are well understood in mathematics and physics. The other $1/f^{\beta}$ noise (Figure 5.1) remains mystery in history but found common in nature.

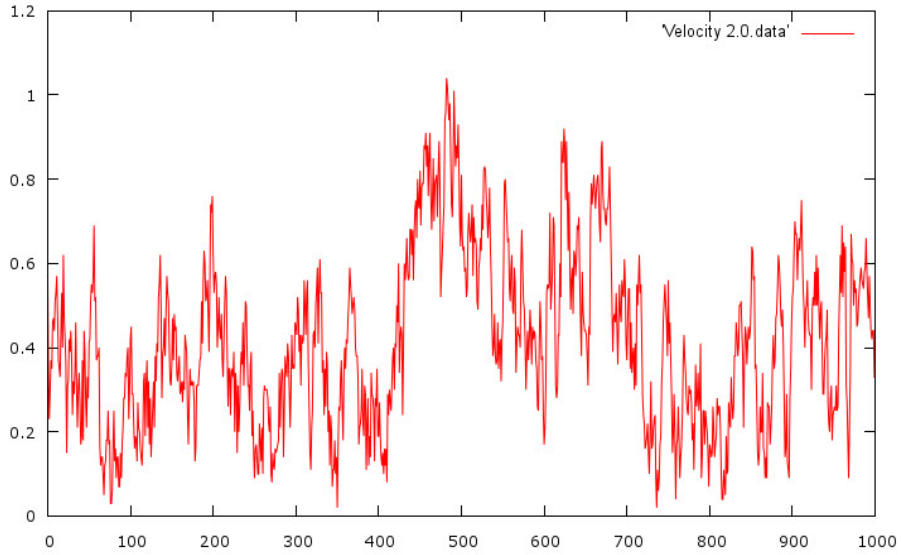
Notice that, in higher dimensions, $S(f)$ is given by

$$S(f_1, ..., f_n) = \frac{1}{(\sqrt{\sum_{i=1}^{n} f_i^2})^{\beta+n-1}} \tag{5.5}$$

where n is the dimensional number.



(a)



(b)

Figure 5.1: fBm representation of wind in this work ($\beta = 5/3$), mean wind velocity U (a) U=4.0m/s (b) U=2.0m/s. The abscissa is time(s), the ordinate is wind speed(m/s).

5.2 Wind field

5.2.1 Wind field implementation

Wind is a complex natural phenomena because of different boundary layers and wind shears. Let the velocity of wind is $V = (V_u, V_v, V_w)$, V_u, V_v, V_w describe the wind velocity components along the x, y and z axes of the coordinate system in the free fall simulation. $U(h)$ is the mean wind velocity at height h. According to the logarithmic wind law [Thuillier and Lappe, 1964], the mean wind velocity is calculated as:

$$U(h) = \frac{u_*}{k} \ln(\frac{h}{z_0}) \tag{5.6}$$

where u_* is the friction velocity (m/s), k is the von Karman's constant ($k = 0.40$). z_0 is the roughness parameter (meter), conceptually it is the height where V goes to 0. h is the height of object above ground (meter). The value of z_0 is different in terms of the types of terrain of the ground (Table 5.1). The fBm method is possible to represent the

Table 5.1: Roughness parameters for different terrains[Ray et al., 2006]

Terrain	Roughness parameter z_0
snow 2	0.003
grass 3	0.008
few trees 4	0.1
many trees, few buildings	0.25
forest	0.5
center of cities with tall buildings	3.0

wind nature [Olesen et al., 1984]. The spectral density function of wind field is given as follows based on Kolmogoroff's law.

$$\frac{nS_u(n)}{u_*^2} = Cf^{-2/3}\phi^{2/3} \tag{5.7}$$

where n is the frequency, f is the reduced frequency, $f = nh/U(h)$. $\phi = \epsilon kh/u_*^3$, ϵ is the dissipation rate according to Kolmogoroff's law. C is a constant, $C = \alpha(2\pi k)^{-2/3}$, α is determined experimentally to be 0.5, so C is 0.3 for spectra in u directions and C=0.4 for v and w wind directions.

Modifying the Equation 5.7 by the definition of $f = nh/U(h)$,

$$S_u(n) = u_*^2(\frac{U(h)\phi}{h})^{2/3}\frac{C}{n^{5/3}} \tag{5.8}$$

In order to get the representation of fBm like $S(f) = A/f^{\beta}$ (A is the amplitude of different wind component direction (u, v, w)), we adopt the same approximations of A_u, A_v, A_w with [O.Khorloo and N.Chiba, 2011].

$$\begin{aligned} A_u &= u_*(\frac{U(h)}{h})^{2/3}, \beta = 5/3 \\ A_v &= 0.88A_u \\ A_w &= 0.55A_u \end{aligned} \tag{5.9}$$

u_* is calculated by Equation 5.6.

To obtain the wind velocities, we apply inverse Fourier transform (Equation 5.1) to Equation 5.5 in the form using FFTW code [Frigo, 1999].

$S_p(f_1, ..., f_n) = \frac{A_p}{(\sqrt{\sum_{i=1}^{n} f_i^2})^{\beta+n-1}}$

Here, p is wind direction (u, v, w).

5.2.2 Implementation results

The results of one-dimensional wind field are shown in Figure 5.1 with the mean wind velocity$U(h) = 4.0m/s$ and $U(h) = 2.0m/s, h = 2.0m, z_0 = 0.3m$.

The results of two-dimensional wind field are shown in Figure 5.2 with the mean wind velocity$U(h) = 4.0m/s$. The grid size in Figure 5.2 is 100×100.

The results of three-dimensional wind field are shown in Figure 5.3 with the mean wind velocity$U(h) = 4.0m/s$. The grid size in Figure 5.3 is $50 \times 50 \times 50$.

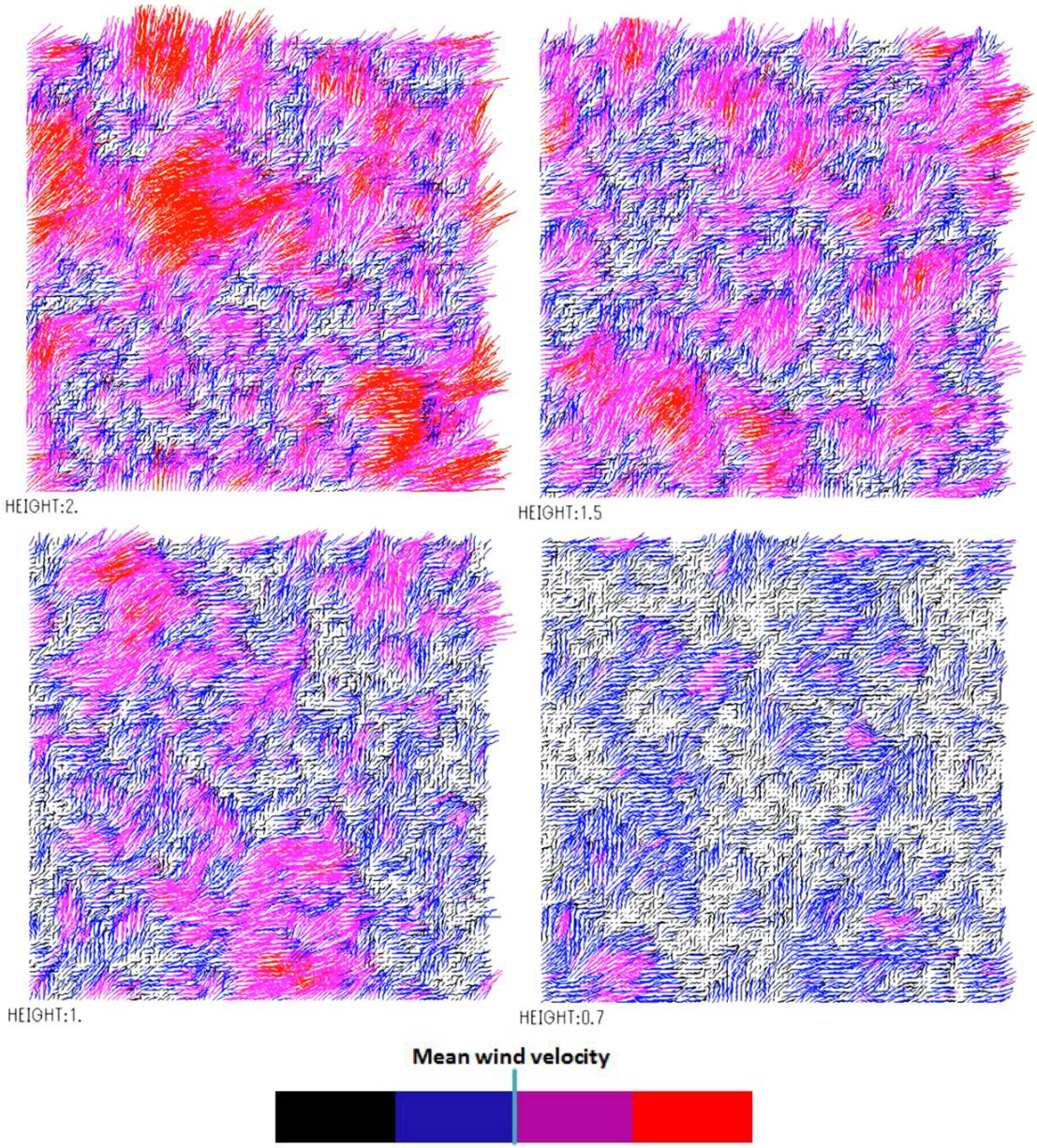


Figure 5.2: Two-dimensional wind field, HEIGHT is the distance (m) above ground. The color used in figure represents velocity length comparing mean wind velocity U(bottom, red: $V > 2U$; pink: $U < V \leq 2U$; blue: $U/2 < V \leq U$; black: $V \leq U/2$), wind direction is along x-axis.

About the computation performance of the wind field implementation, the computation cost of two-dimensional wind field is shown in Table 5.2.

Table 5.2: Computation cost of 2D wind field

100×100	500×500	1000×1000	2000×2000
8ms	129ms	542ms	2221ms

The computation cost of three-dimensional wind field is shown in Table 5.3.

Table 5.3: Computation cost of 3D wind field

$50 \times 50 \times 50$	$70 \times 70 \times 70$	$100 \times 100 \times 100$	$150 \times 150 \times 150$
170ms	473ms	1363ms	4586ms

In order to achieve accurate real-time free fall simulation in a wind field, we prefer to choose the two-dimensional wind field calculation. For the wind-field of different heights, the wind velocity relationship is obtained from Equation 5.6.

$$\frac{V(h_1)}{V(h_2)} \approx \frac{U(h_1)}{U(h_2)} = \frac{\ln(h_1) - \ln(z_0)}{\ln(h_2) - \ln(z_0)} \tag{5.10}$$

In Figure 5.2, the wind fields of different heights (Height=2.0m, 1.5m, 1.0m, 0.7m) are calculated using Equation 5.10, which are based on the 2D wind field of $h = 2.0m$.

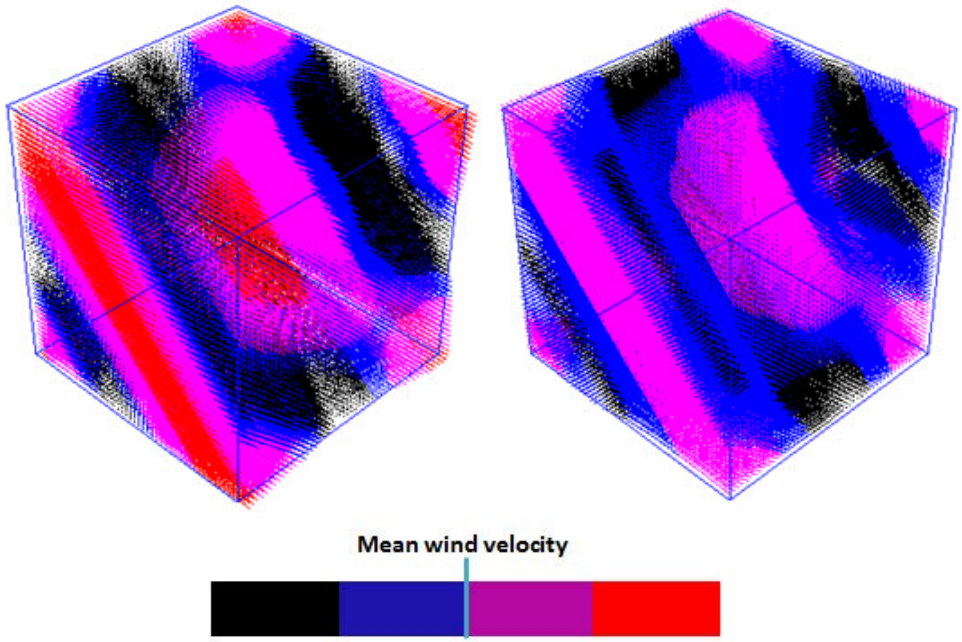


Figure 5.3: Three-dimensional wind field, the color used in figure represents velocity length comparing mean wind velocity U(bottom, red: $V > 2U$; pink: $U < V \leq 2U$; blue: $U/2 < V \leq U$; black: $V \leq U/2$)

5.3 Wind-object interaction

The external forces when an object freely falling in wind are mainly wind force and collision among the objects. In the motion synthesis of free fall in still flow (Chapter 4) , the influence of drag, lift, gravity forces are already considered. Furthermore, the frequency of collisions among falling objects is fairly low by the observations from experiments, so that we neglect the influences of collisions. In this section, the wind-object interaction focuses on tackling wind field into synthesized trajectory.

Let set $u(p, h, t)$ is a 2D wind field at height position p and h (Equation 5.10), the wind velocity at point $p_0 = (x_0, y_0, z_0)$ is $u(p_0, z_0, t_0) = (u_x, u_y, u_z)$ (u_x, u_y, u_z are corresponding to the u,v,w components in Equation 5.9).

About the computed trajectory of falling objects in still flow, we set the curve to be a function $f(t)$, $f(t)$ is a set of points per frame in time domain. At time t_0, $f(t_0)$ is a quaternion (p_0, θ_0) about position and orientation.

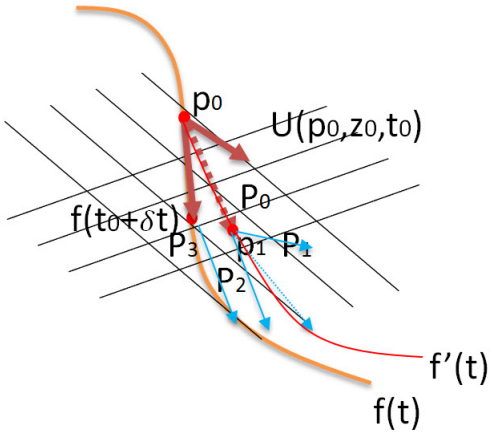


Figure 5.4: Wind-object interaction

After time step δt, $f(t_0 + \delta t)$ is at point p'. The next point after p_0 is set to be at p_1, $\widehat{p_0p_1} = u(\widehat{p_0, z_0}, t_0) + \widehat{p_0p'}$ (Figure 5.4). If p_1 does not coincide with any grid node of wind field, assuming the neighbouring 2D grid nodes around p_1 are $Pi(0 \leq i \leq 3)$, the wind velocity at p_1 is calculated by the linear interpolation of u_i at P_i. After the iterations, new trajectory $f'(t)$ of falling object in wind is synthesized.

About the rotation under the influence of wind, we notice that falling object(like leaf and paper pieces) is subjecting to change angle in a wind field. To achieve the realistic effects, we apply the noise function into the orientation calculation (Equation 5.11).

$$\theta(t) = WN(t) \tag{5.11}$$

where $N(t) = \sqrt{u_x^2 + u_y^2 + u_z^2}$ is from the fBm noise function at Section 5.2.1, and W is the maximum motion angle, which is designated by animator. The rotation axes of the falling objects is obtained by looking up the Figure 4.6.

Chapter 6

Experimental Results

This chapter verifies the Hypothesis 4.1 by the experiments, the simulation results by our approaches give a realistic and natural simulation in real-time environment.

6.1 Experiment setup

126 simple experiments of free falling objects have been done for verifying the hypothesis 4.1 in Section 4.1. we record the results of primitive motions $\{L_i\}$, for example, the motion transits from L_1 to L_6 in the falling processes.

These experiments are executed by an experimenter releasing the varied objects from hand in air (Figure 6.1), including three ellipse papers (a=2cm, L=5, 6, 8cm), three round papers (radius: 2cm, 3.5cm, 5cm), two square papers(a=1, 2cm), three rectangle papers(a=1cm, L=3, 5, 8cm), two leaves(L=5, 8cm). The motions are captured by high speed camera Casio's EX-F1, which can shoot video in 300fps, 600fps and up to 1200fps.

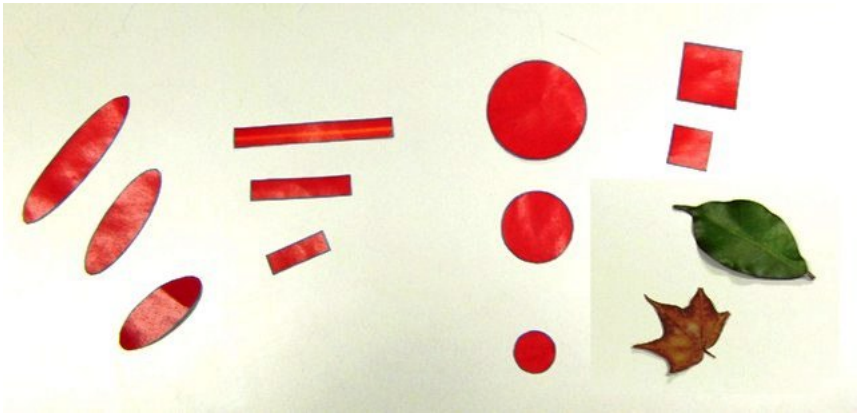


Figure 6.1: Released objects in experiments

We find that it is difficult to capture the free fall motion because some trajectories have long distance far away from the camera. Because of this, about one-fourth of the experiments failed. The following motion classes are clarified from the observation of captured videos using the label $\{L_i | 1 \leq i \leq 6\}$.

Table 6.1: Motion classes table from simple experiments

Class 1	$L_2 \rightarrow L_5, L_2 \rightarrow L_6$
Class 2	L_2
Class 3	$L_1 \rightarrow L_4$
Class 4	$L_3 \rightarrow L_6$, L_3
Class 5	$L_2 \rightarrow L_6$
Class 6	L_4, L_5

It is obvious that all the classes in Table 5.1 are content with the hypothesis. And the classes are similar to the motion classes Table 4.1 by [Razavi, 2010]. Figure 6.2, Figure 6.3 and Figure 6.4 are experimental results. Notice that the videos captured by high speed shoot are low-resolution, it is not suitable for image processing.

Figure 6.2: Sequential snapshots of releasing paper in air

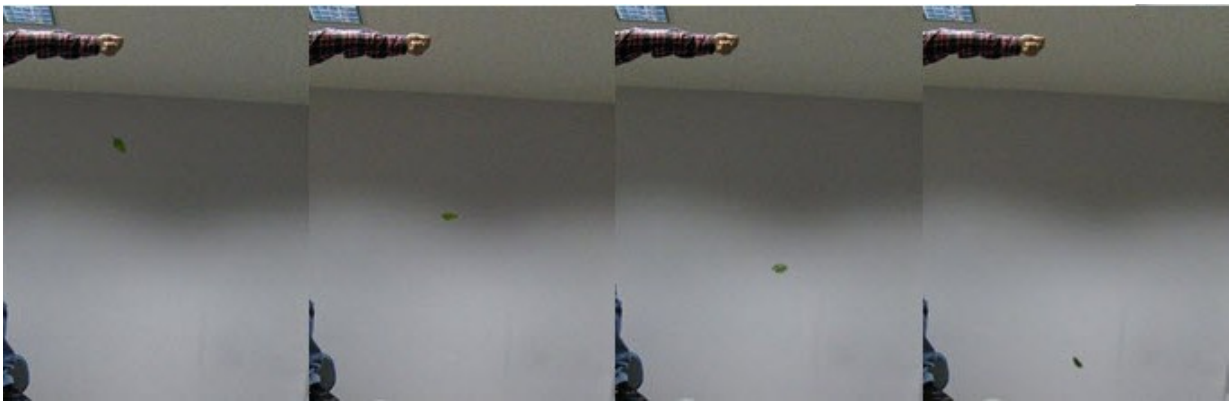


Figure 6.3: Sequential snapshots of released leaf in air

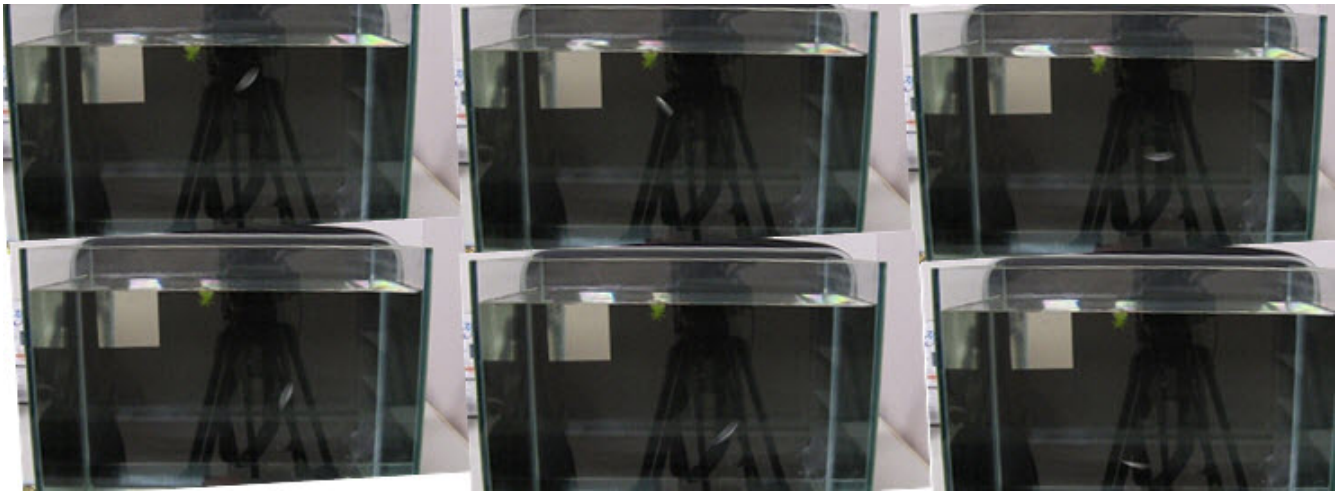


Figure 6.4: Sequential snapshots of released coin in water

6.2 Simulation results

From input parameters as shown in Section 2.4, the main primitive motion L_i is defined by looking up phase diagram using the calculated I^* and Re. According to the Hypothesis 4.1, the global path synthesis starts from a random primitive motion $L_j(i < j)$, and transformed to the next primitive motion which are estimated by Markov model in Section 4.2. The amplitudes of primitive motions are solved in Subsection 4.4.1 and the synthesis of the primitive motion are selected from trajectory database in Chapter 3. About the evaluation of this work, comparison with ground truth from video is a direct and efficient way. Although the limit of chaotic motions exists, the characteristics of free fall motion are well captured in this simulation satisfying observations and theoretical analysis.

As simulation results, we supply three cases of free falling motions as shown in Table 6.2.

Table 6.2: Attributes of the falling objects

object	shape	fluid	size	Re	I^*
aluminium	circle	water	$a = 2.0cm, b = 0.15cm$	3.55×10^3	10^{-2}
leaf	ellipse	air	$L = 7.3cm, a = 4.2cm, b = 0.03cm$	1.2×10^4	6.3×10^{-3}
paper	ellipse	air	$L = 8.0cm, a = 2.0cm, b = 0.01cm$	6.8×10^3	2.2×10^{-3}

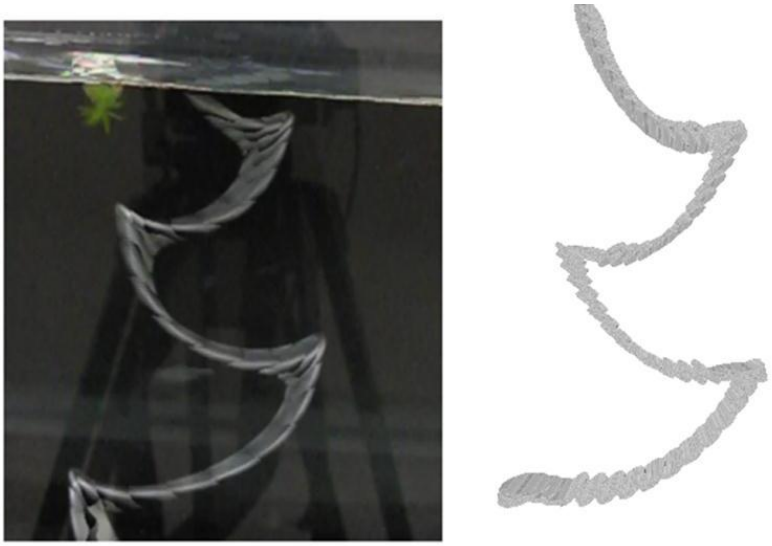


Figure 6.5: Comparison of our simulation with the real free fall of one Japanese yen coin in water

While the example shown in Figure 6.5 suggests that our result is realistic, it is variable for not only air but all fluids where the object fall freely. As shown in the left image, an aluminium circular disk had fallen in the water from 50 cm height. A regular fluttering motions are observed. According the initial parameter in Table 1, $I^* = 10^{-2}$, and $Re = 3.55 \times 10^3$. By looking up the phase diagram, the main motion prototype is fluttering (PF). Then using the free fall motion graph and Markov chain model, we get the positions and orientations from motion sequences constituted by pre-computed database, and the result is shown in the right figure.

Figure 6.6 shows a leaf falls in air from 200cm height, the perpendicular falling, tumbling and helix motions are found here. We calculated $I^* = 6.3 \times 10^{-3}$ and $Re = 1.2 \times 10^4$, so the main motion is transitional helix motion. The connection of three primitive motions observed is well fit in our model. Between the connections among each motion sequences, the segment interpolation result is presented in the right figure.

Figure 6.7 shows a red paper falling in air, and $I^* = 2.2 \times 10^{-3}$ and $Re = 6.8 \times 10^3$, which is approximately spiral motion in the phase diagram. And the real motion is segmented into tumbling and spiral motions, which have verified the hypothesis in Chapter 4. According to the orientation modification in optimization, the falling object fits type

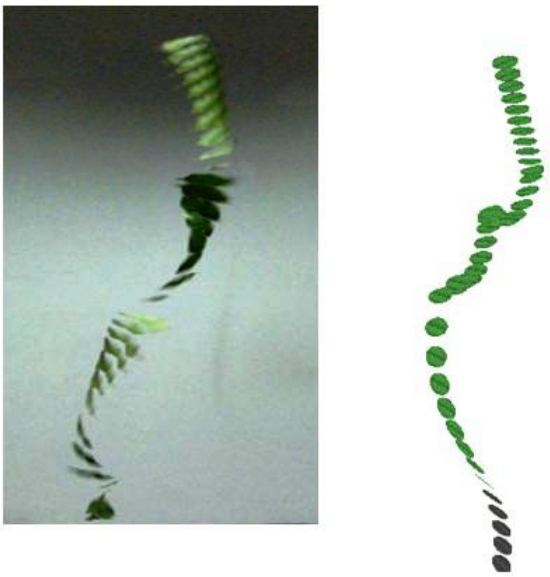


Figure 6.6: Comparison of our simulation rslut with the real free fall of a leaf released from 2.0m height

B and the simulation is flutter in length and rotation in length axes.

In Figure 6.11, the free fall motion paths are coupled with different wind fields. The black dots in the figure are the control points for Bezier curves synthesis. (a) is in a low wind field, so the motion paths is similar with the reference in Figure 6.9. In (b) and (c) conditions, the influences of wind are apparent.

Figure 6.12 shows the simulation results of the free fall motion in two types of wind fields, which are corresponding to the conditions of Figure 6.11(b) and (c). We found that the tumbling motion disappears under wind field and the object goes far away.

All examples (Figure 6.8, Figure 6.9, Figure 6.10, Figure 6.11, Figure 6.12) were implemented in C++ on an Intel Core i7 CPU 3.20 GHz and 12.0 GB RAM in real time (around 50fps). Because most parts of our method are executed off-line, the on-line motion synthesis and optimization process are rarely memory consuming, so our simulation is realistic and predictable, so it is feasible for interactive applications.

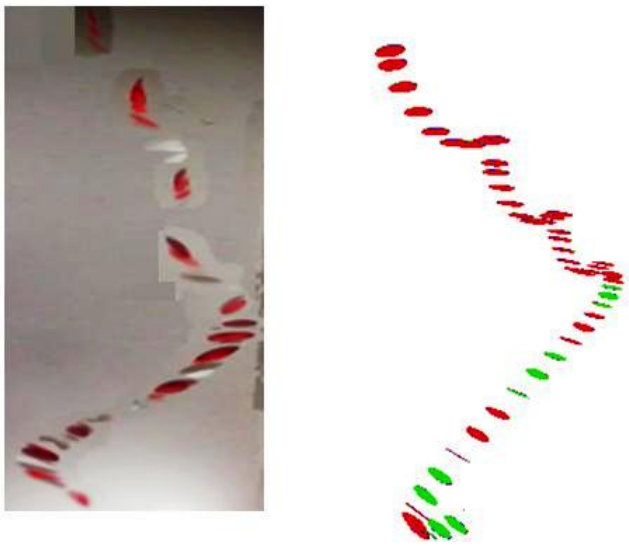


Figure 6.7: Comparison of our simulation with the real free fall of a red ellipse paper in air released from 3.1m height

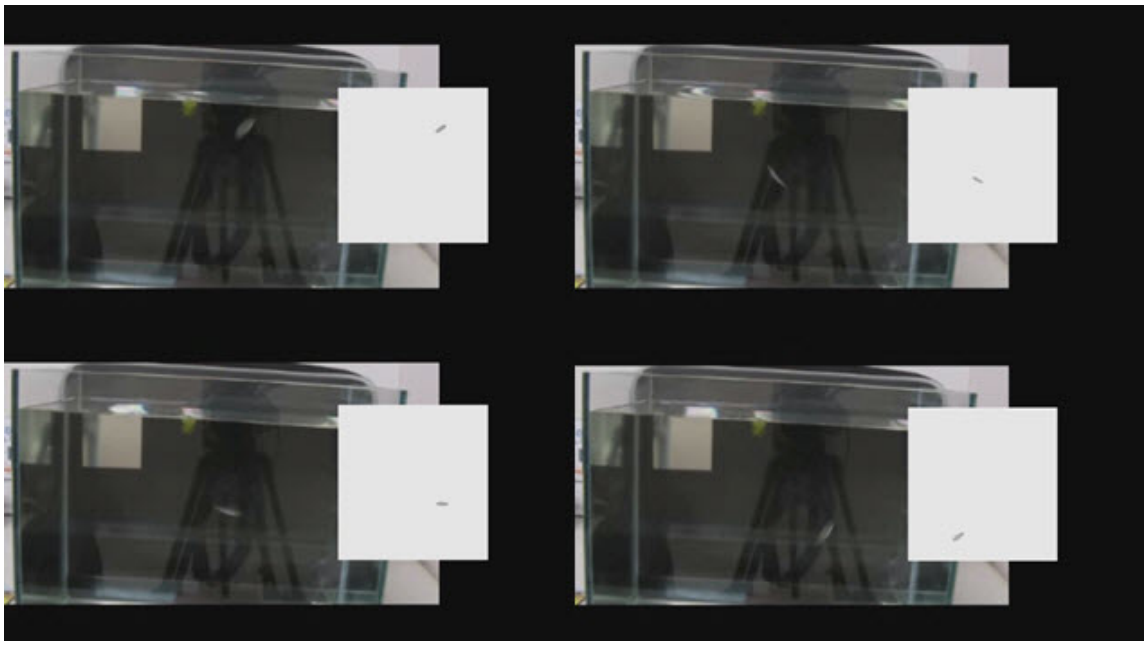
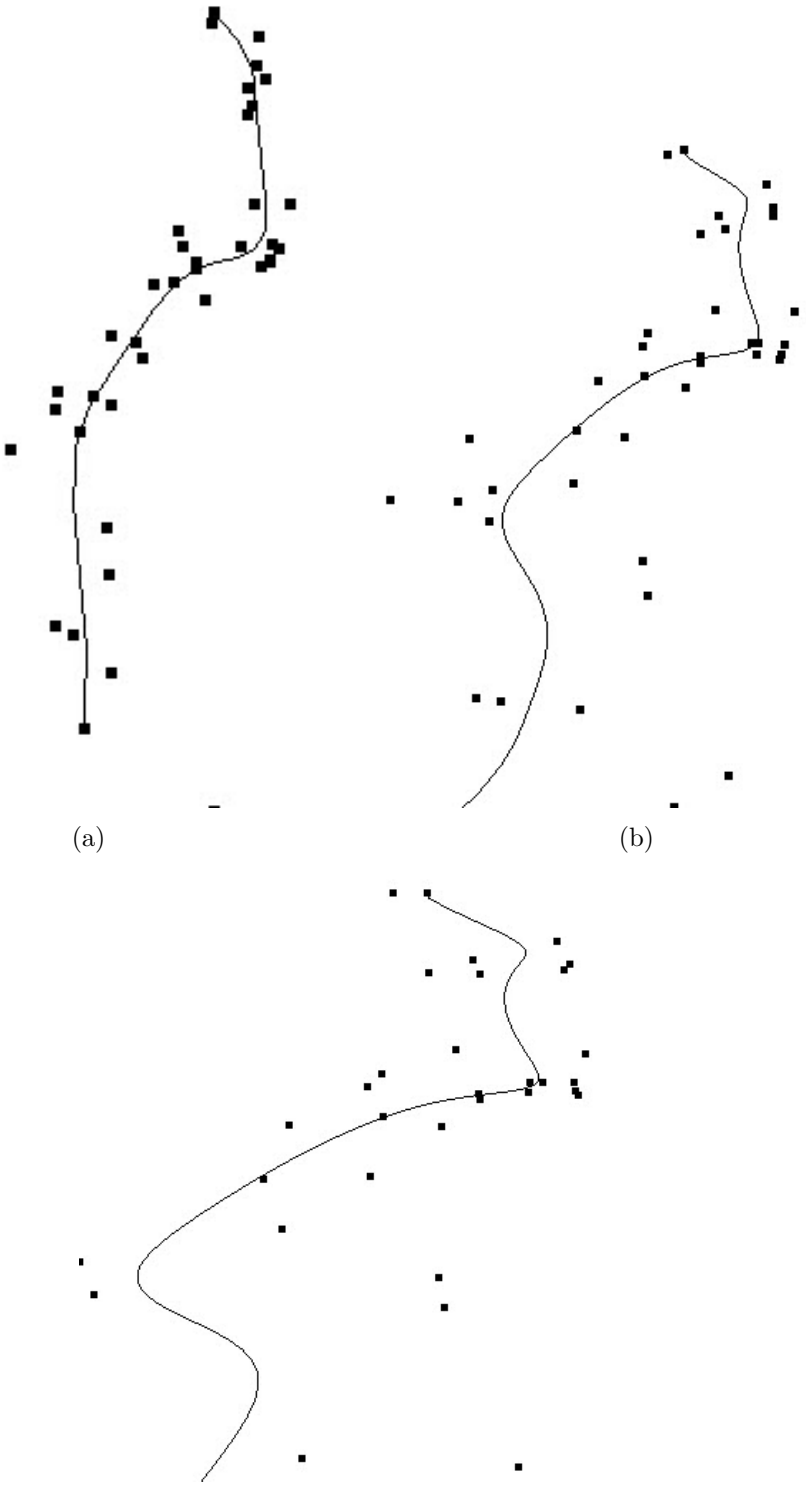


Figure 6.8: Comparison of our simulation result with video for falling coin

Figure 6.9: Comparison of our simulation result with video for falling leaf

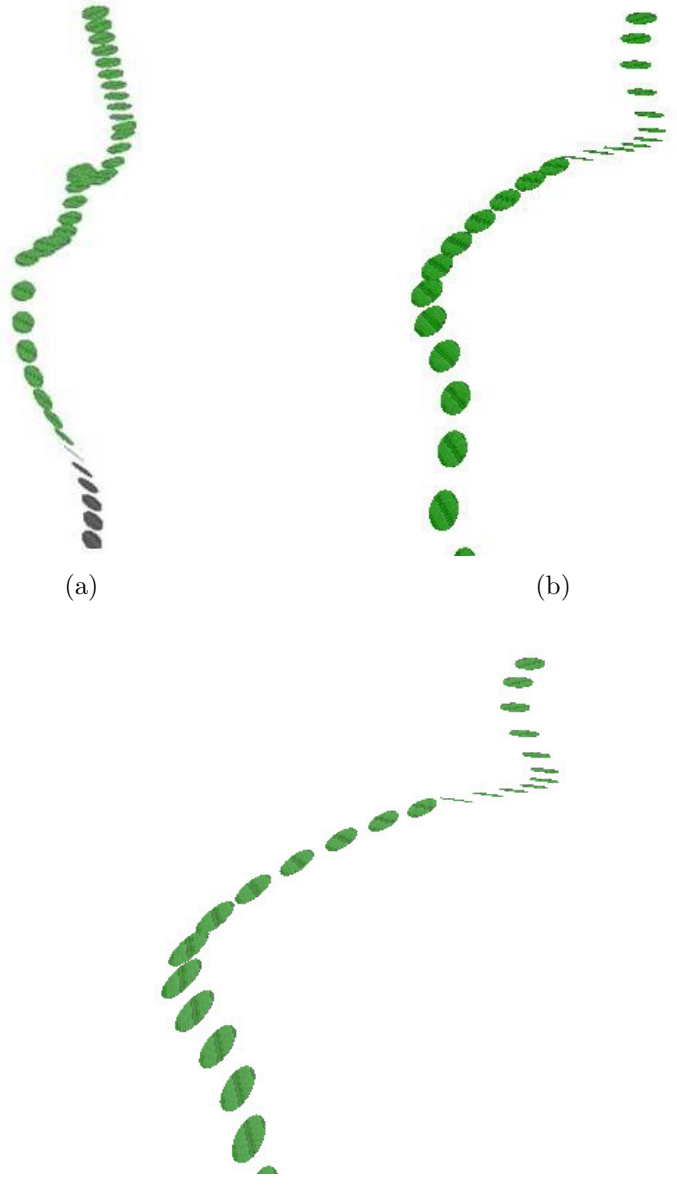
Figure 6.10: Comparison of our simulation result with video for falling paper



(a) (b)

(c)

Figure 6.11: Free fall motion paths in different wind field, mean wind velocity U(a): $U = 1.0m/s$;(b): $U = 3.0m/s$ (c): $U = 5.0m/s$). Wind direction is from right to left.



(a) (b)

(c)

Figure 6.12: Final synthesized free fall motion in different wind field. (a) no wind as reference; (b) mean wind velocity is 3.0m/s; (c) 5.0m/s. Wind direction is from right to left.

Chapter 7

Summary

7.1 Research conclusion

This work presents a framework for generating free fall motion by data-driven motion synthesis and precomputed trajectory database. This research is the first research about the physical details of free fall motion and proposed an efficient motion synthesis approach to achieve realistic and predictable free fall simulation.

This work tackles regular geometries, including rectangular, circular, and elliptical objects with constant densities. For irregular geometry and uneven density distribution objects, it is complicated to consider the influence to the motion by the geometry and density distribution modifications. When a paper or plastic object falls freely, the change of shape also happens during the motion, which we omitted in the framework.

7.2 Future directions

This work adopted the data-driven approach and pre-computed method, and combined the ODEs and example data from experiments. It is a significant challenge in computer graphics to fill the gap between physically-based simulation and example-based animation. The important issue is to create a new realistic and low-cost model for visual simulations.

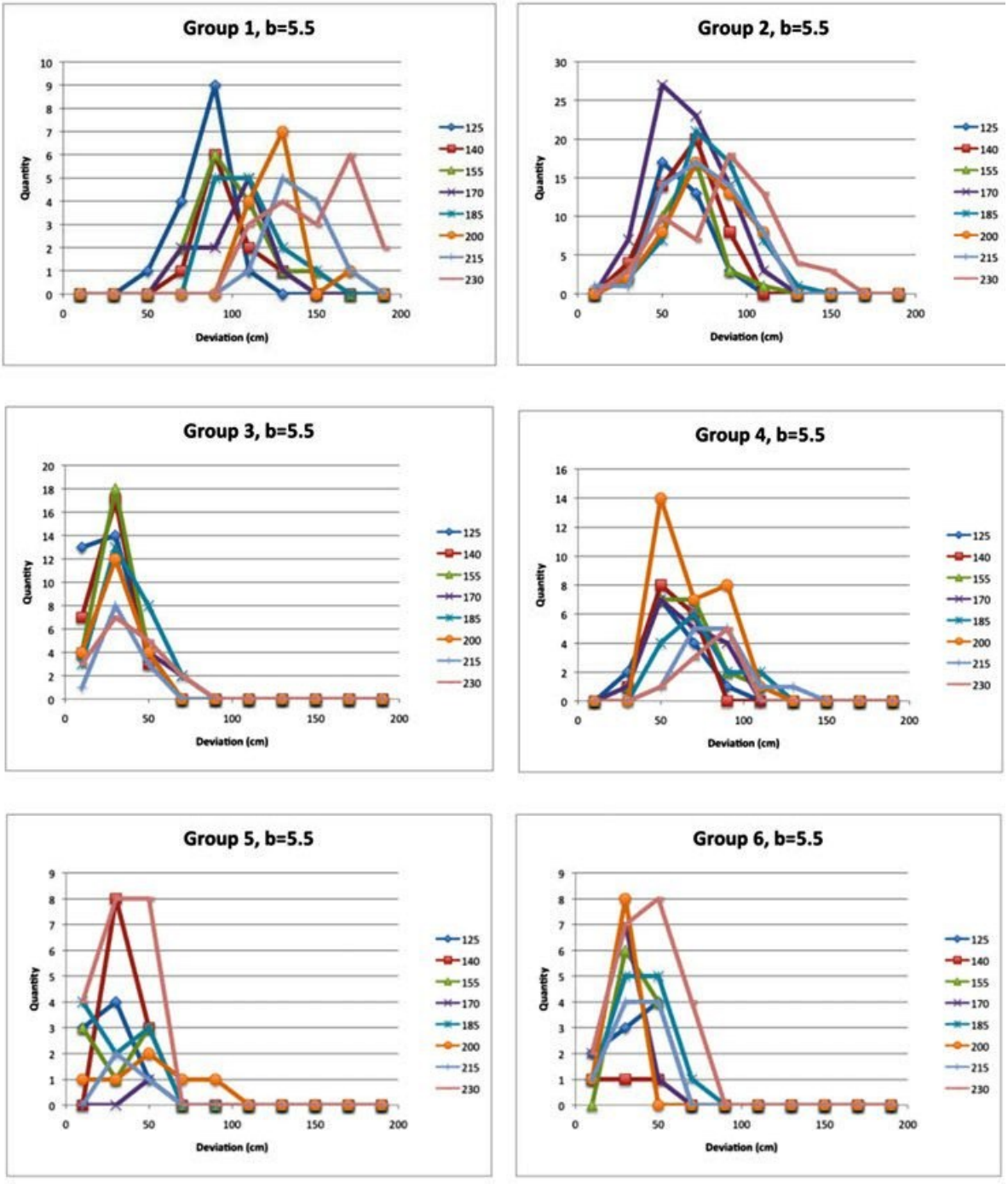
In the future, it is possible to imply the current simulation results as patterns in fluid simulation by pattern based method [Yuan et al., 2011]. Because the free fall is a common phenomena for light-weight object, it will be promising to coupling the simulation into game industry and other simulations in computer graphics.

The simulation of the oscillations is an intriguing topic which could happen in different directions while the object has high degree-of-freedom. The quantitative analysis of the oscillation in free fall will be another feasible approach of motion synthesis.

Appendix

notation	meaning
$A_\perp$	drag coefficient in perpendicular direction
$A_\parallel$	drag coefficient in parallel direction
A_θ	drag coefficient of angular velocity
A_x	amplitude of horizontal velocity
A_y	amplitude of vertical velocity
A_e	amplitude of elliptical oscillation
ϵ_e	aspect ratio of elliptical oscillation
Re	Reynolds number
h	release height
L	length of falling object
a	width of falling object
b	thickness
ρ_s	density of falling object
ρ_f	density of flow
ν	kinematic viscosity of flow
g	gravity acceleration
z_0	roughness parameter of terrain
Fr	Froude number
L	lift force
ω	angular frequency
x	x position
y	y position
z	z position
m	mass
V	volume of falling object
θ	angle of falling object
u	velocity in x direction
v	velocity in y direction
I	moment of inertia
I^*	dimensionless moment of inertia
U	velocity of flow
ϵ	aspect ratio
p	pressure
k	von Karman's constant 0.4

Spatial derivations of motion groups [Razavi, 2010]



Publication

[1] **Xie, H.**, and Miyata, K. (2011). Pre-computed Data-driven Free Fall Animation. *NICOGRAPH International 2011*, Fluid Simulation 3-5, Kanagawa, Japan.

[2] **Xie, H.**, and Miyata, K. (2011). Free fall motion synthesis. *In SIGGRAPH Asia 2011 Sketches (SA '11)*, ACM, New York, NY, USA, Article 6.

Submitting paper

[3] **Xie, H.**, and Miyata, K. (2012). Realistic Motion Synthesis of Free Falling Behaviours. *Proceedings of Computer Graphics International 2012*, Bournemouth, UK.

Other Work

[4] Vincent de Paul ATANGANA EKANI, **Haoran Xie**, Kazunori Miyata, Dancing Sound, *The Institute of Image Information and Television Engineers Technical Report*, Vol.34, No.44, pp.21-24, 2010

Bibliography

[Andersen et al., 2005] Andersen, A., Pesavento, U., and Wang, Z. J. (2005). Unsteady aerodynamics of fluttering and tumbling plates. *Journal of Fluid Mechanics*, 541:65–90.

[Aoki et al., 2004] Aoki, K., Hasegawa, O., and Nagahashi, H. (2004). Behavior learning and animation synthesis of falling flat objects. *JACIII*, 8(2):223–230.

[Arikan and Forsyth, 2002] Arikan, O. and Forsyth, D. A. (2002). Interactive motion generation from examples. *ACM Trans. Graph.*, 21:483–490.

[Belmonte et al., 1998] Belmonte, A., Eisenberg, H., and Moses, E. (1998). From flutter to tumble: Inertial drag and froude similarity in falling paper. *Physical Review Letters*, 81:345–348.

[Bruce R. Munson, 2001] Bruce R. Munson, Donald F. Young, T. H. O. (2001). *Fundamentals of Fluid Mechanics*. Wiley; 4 edition.

[Desbenoit et al., 2006] Desbenoit, B., Galin, E., Akkouche, S., and Grosjean, J. (2006). Modeling autumn sceneries. In *Eurographics 2006, short Papers.*

[Field et al., 1997] Field, S. B., Klaus, M., Moore, M. G., and Nori, F. (1997). Chaotic dynamics of falling disks. *nature*, 388:252–254.

[Foley, 2000] Foley, J. (2000). Getting there: The ten top problems left. *IEEE Comput. Graph. Appl.*, 20:66–68.

[Frigo, 1999] Frigo, M. (1999). A fast fourier transform compiler. *SIGPLAN Not.*, 34:169–180.

[Garg and Nayar, 2006] Garg, K. and Nayar, S. K. (2006). Photorealistic rendering of rain streaks. *ACM Trans. Graph.*, 25:996–1002.

[Haevre et al., 2006] Haevre, W. V., Fiore, F. D., and Reeth, F. V. (2006). Physically-based driven tree animations. In Chiba, N. and Galin, E., editors, *NPH*, pages 75–82. Eurographics Association.

[Haiyan et al., 2010] Haiyan, L., Qingjie, S., Hui, Z., and Qin, Z. (2010). Example-based motion generation of falling leaf. In *Computer Design and Applications (ICCDA), 2010 International Conference on*, volume 5, pages V5–217 –V5–221.

[Hong and Kim, 2003] Hong, J.-M. and Kim, C.-H. (2003). Animation of bubbles in liquid. *Comput. Graph. Forum*, 22(3):253–262.

[Kijima, 1997] Kijima, M. (1997). *Markov Processes for Stochastic Modeling*. Chapman and Hall/CRC.

[Kolomenskiy and Schneider, 2010] Kolomenskiy, D. and Schneider, K. (2010). Numerical simulations of falling leaves using a pseudo-spectral method with volume penalization. *Theoretical and Computational Fluid Dynamics*, 24:169–173. 10.1007/s00162-009-0171-0.

[Koumoutsakos et al., 2008] Koumoutsakos, P., Cottet, G.-H., and Rossinelli, D. (2008). Flow simulations using particles: bridging computer graphics and cfd. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 25:1–25:73, New York, NY, USA. ACM.

[Kovar et al., 2002] Kovar, L., Gleicher, M., and Pighin, F. (2002). Motion graphs. *ACM Trans. Graph.*, 21:473–482.

[Maxwell, 1854] Maxwell, J. C. (1854). On a particular case of the descent of a heavy body in a resisting medium. *Camb. and Dublin Math.J.*, 9:145–148.

[Mei et al., 2007] Mei, X., Decaudin, P., and Hu, B.-G. (2007). Fast hydraulic erosion simulation and visualization on gpu. In *Pacific Conference on Computer Graphics and Applications*, pages 47–56.

[Mount, 2005] Mount, D. (2005). Kmlocal: A testbed for k-means clustering algorithms. Available at http://www.cs.umd.edu/ mount/Projects/KMeans/km-localdoc.pdf.

[Müller et al., 2003] Müller, M., Charypar, D., and Gross, M. (2003). Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 154–159, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Narain et al., 2010] Narain, R., Golas, A., and Lin, M. C. (2010). Free-flowing granular materials with two-way solid coupling. *ACM Trans. Graph.*, 29:173:1–173:10.

[Newton, 1687] Newton, I. (1687). *Mathematical Principles of Natural Philosophy*. transl. B. Cohen & A. Whitman, University of California Press, this edition published 1999, Berkeley, USA.

[Okamoto et al., 2009] Okamoto, T., Fujisawa, M., and Miura, K. T. (2009). An interactive simulation system for flying japanese kites. In *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, Sandbox '09, pages 47–53, New York, NY, USA. ACM.

[O.Khorloo and N.Chiba, 2011] O.Khorloo, Z.Gunjee, B. S. and N.Chiba (2011). Wind field synthesis for animating wind-induced vibration. *The International Journal of Virtual Reality*, 10:53–60.

[Olesen et al., 1984] Olesen, H. R., Larsen, S. E., and Hjstrup, J. (1984). Modelling velocity spectra in the lower part of the planetary boundary layer. *Boundary-Layer Meteorology*, 29:285–312. 10.1007/BF00119794.

[Ota et al., 2004] Ota, S., Tamura, M., Fujimoto, T., Muraoka, K., and Chiba, N. (2004). A hybrid method for real-time animation of trees swaying in wind fields. *The Visual Computer*, 20:613–623. 10.1007/s00371-004-0266-y.

[Park and Kim, 2005] Park, S. I. and Kim, M. J. (2005). Vortex fluid for gaseous phenomena. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '05, pages 261–270, New York, NY, USA. ACM.

[Peitgen and Saupe, 1988] Peitgen, H.-O. and Saupe, D., editors (1988). *The Science of Fractal Images*. Springer-Verlag New York, Inc., New York, NY, USA.

[Ray et al., 2006] Ray, M. L., Rogers, A. L., and Mcgowan, J. G. (2006). Analysis of wind shear models and trends in different terrains. *Renewable Energy*, pages 1–14.

[Razavi, 2010] Razavi, P. (2010). On the motion of falling leaves. *ArXiv e-prints*.

[Reissell and Pai, 2001] Reissell, L.-M. and Pai, D. K. (2001). Modeling stochastic dynamical systems for interactive simulation. *Comput. Graph. Forum*, 20(3).

[Shi et al., 2005] Shi, L., Yu, Y., Wojtan, C., and Chenney, S. (2005). Controllable motion synthesis in a gaseous medium. *The Visual Computer*, 21(7):474–487.

[Shinya and Fournier, 1992] Shinya, M. and Fournier, A. (1992). Stochastic motionmotion under the influence of wind. *Computer Graphics Forum*, 11(3):119–128.

[Stam, 1999] Stam, J. (1999). Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 121–128, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[Stam and Eugene, 1993] Stam, J. and Eugene, F. (1993). Turbulent wind fields for gaseous phenomena. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 369–376, New York, NY, USA. ACM.

[STRINGHAM and GUY, 1969] STRINGHAM, G.E., S. D. B. and GUY, H. P. (1969). The behavior of large particles falling in quiescent liquids. *GEOL SURV PROF PAP 562-C*, 562:C1–C36.

[Tanabe and Kaneko, 1994] Tanabe, Y. and Kaneko, K. (1994). Behavior of a falling paper. *Phys. Rev. Lett.*, 73:1372–1375.

[Thuillier and Lappe, 1964] Thuillier, R. H. and Lappe, U. O. (1964). Wind and temperature profile characteristics from observations on a 1400 ft tower. *Journal of Applied Meteorology*, 3:299–306.

[Tritton, 1988] Tritton, D. J. (1988). *Physical Fluid Dynamics, Second Edition*. Oxford Science Publications.

[Vázquez and Balsa, 2008] Vázquez, P.-P. and Balsa, M. (2008). Rendering falling leaves on graphics hardware. *Journal of Virtual Reality and Broadcasting*, 5(2).

[Wei et al., 2003] Wei, X., Zhao, Y., Fan, Z., Li, W., Yoakum-Stover, S., and Kaufman, A. (2003). Blowing in the wind. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 75–85, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Weissmann and Pinkall, 2010] Weissmann, S. and Pinkall, U. (2010). Filament-based smoke with vortex shedding and variational reconnection. *ACM Trans. Graph.*, 29:115:1–115:12.

[Willmarth et al., 1964] Willmarth, W. W., Hawk, N. E., and Harvey, R. L. (1964). Steady and Unsteady Motions and Wakes of Freely Falling Disks. *Physics of Fluids*, 7:197–208.

[Yu and Turk, 2010] Yu, J. and Turk, G. (2010). Reconstructing surfaces of particle-based fluids using anisotropic kernels. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, pages 217–225, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Yuan et al., 2011] Yuan, Z., Chen, F., and Zhao, Y. (2011). Pattern-guided smoke animation with lagrangian coherent structure. *ACM Trans. Graph.*, 30:136:1–136:8.

[Zhang et al., 2007] Zhang, L., Zhang, Y., Jiang, Z., Li, L., Chen, W., and Peng, Q. (2007). Precomputing data-driven tree animation. *Journal of Visualization and Computer Animation*, 18(4-5):371–382.

[Zhong et al., 2011] Zhong, H., Chen, S., and Lee, C. (2011). Experimental study of freely falling thin disks: Transition from planar zigzag to spiral. *Physics of Fluids*, 23(1):011702.