Physically-Based Animation in Performing Accuracy Bouncing Simulation

Loh Ngiik Hoon[1]
Faculty of Applied and Creative Arts
Universiti Malaysia Sarawak
Kota Samarahan, Sarawak
Malaysia

***Abstract*—This study investigates the use of physics formulas in achieving plausible bouncing simulation in animation. The need for physics animation was to produce visually believable animations adhering to the basic laws of physics. Based on the review, the creation of accurate timing simulation in bouncing dynamic was significantly difficult particularly in setting keyframes. It was comprehensible that setting the value of keyframes was unambiguous while specifying the timing for keyframes was a harder task and often time-consuming. The case study of bouncing balls' simulation was carried out in this research and the variables of mass, velocity, acceleration, force, and gravity are taking into consideration in the motion. However, the bouncing dynamic is a significant study in animation. It is often used and it shows many different aspects of animations, such as a falling object, walking, running, hopping, and juggling. Therefore, the physical framework was proposed in this study based on numerical simulations, as the real-time animation can be addressed for controlling the motion of bouncing dynamic object in animation. Animation based physics algorithm provided the animator the ability to control the realism of animation without setting the keyframe manually, to provide an extra layer of visually convincing simulation.**

***Keywords*—Bouncing simulation; physics algorithm; physics animation; real time animation; animation**

I. Introduction

The concept of physically based in animation has been long established by Disney artists through "The Twelve Basic Principles of Animation". The main purpose of the twelve basic principles was to produce more realistic animations adhering to the basic laws of physics [1]. In the context of 3D computer animation, realistic timing is extremely important to add a life-like quality to animate objects and give the animation some real-world authority. The proper timing is crucial to make the ideas readable. Consequently, animation artists carefully study the motion of the objects by adding quality and accuracy to generate realistic-looking animations. Thus, the concept of applying the laws of physics in animation has further gained importance to generate an accurate timing animation, and there is a need for consideration of physics motion in this field.

The concept of physics motion can be interpreted with the principle of bouncing ball simulation. The bouncing ball is the most basic and one of the most important animation exercises. Bouncing ball simulation causes the natural factors of motion, velocity, acceleration, mass, gravity, friction, elasticity, or squash and stretch, and timing. Hence, this concept of a bouncing ball is often used in the animation as a reference because it integrates several fundamental concepts that animators apply to just about everything they animate. The author in [2] also explained, bouncing ball is a common model for numerous rhythmic tasks such as walking, running, hopping, and juggling, and it has been an extensive study which provides a theoretical basis for control of such rhythmic tasks in animation. Last but not least, realistic bouncing ball simulation showed the significance in the animation that a lot of physical measurements are required. Plus, timing an animation is often the most difficult part to set the spatial values of the keyframe in achieving realistic simulation. Most of the users are unable to imagine the timing and convey it using the provided interfaces. Therefore, the physics-based approach is a well-adapted concept to simulate believable animations. Based on the physical motion regarding numerical simulations framework, the animator is provided with the ability to control the realism motion of simulated object without setting the keyframe manually, by adding an extra layer of visually convincing animation. Hence, the different section of introduction, literature review, analysis physics motion of bouncing simulation, physics motion, real time dynamic bouncing ball, result and findings, implication, discussion, and conclusion are discussed in this paper.

II. Literature Review

A. Concept of Physically based Animation

Physically-based animation has emerged as a core area of computer graphics finding widespread application in the film and video game industries as well as in areas such as virtual surgery, virtual reality, and training simulations [3]. With the advance production technology, it allows designers to create animation by their own will with the greatest degree of freedom, but the products of the technology are not very good in terms of natural performance [4]. In order to fit into the current trend of fast increase of computer processing and user experience, physically-based animation is a well-adapted concept to simulate the realistic-looking animations with self-controllable performance. According to [5], the physics-based approach uses the law of physics to simulate motion and interaction with the environment. In his study, he pointed out a complete and effective system for animation should integrate key-framing and physics-based techniques. Key-framing allows objects or characters to perform unnatural tasks.

However, physics-based simulation models the object or character's interaction with the environment in a physical way and ensuring a realistic result. The author in [6] also described physical dynamics are based on two basic notions, which are material points and forces that induce movement. In the study, researchers explained clearly about these two notions that are included in their model and present a first simulation algorithm. The author in [7] also emphasized that physically-based models are well suited to simulate natural motion and flexible elastic objects. Moreover, [8] emphasized that physically-based animation tends to model only perfect worlds. Modeling a completely realistic physical world needs to talk in all external factors, including shifting winds, air humidity, different materials, and attraction forces from every object. All these factors will affect the moving objects in the real world. The study also showed that the existence of physics-based animation and alternative algorithms is a way to allow the animator to manipulate a rigid body during the simulation and have the computer to make the necessary adjustment to position and velocity. The results are presented in a good visual-oriented part and believability through several performances. From the findings, the researcher suggests that the animation developed with the reference to physics-based simulation to give better performance and even better visual results. The results also proved the inaccuracy will be minimal for a visual result, and thus the believability is retained by using the physically-based method.

B. Bouncing Ball Animation

Based on [9], bouncing ball simulation is the standard animation test for all the beginners. It is one of the most important subjects to learn animation and is essential to know as an animator. From this simple simulation, principles of animation such as timing, squash and stretch, arcs, volume, and weight can be learned. The concept of the bouncing ball is shown in Fig. 1. The ball travels up and down through space while travels horizontally from left to right. With each bounce, the ball loses height because friction with the ground reduces the momentum of the initial force that set the ball in motion. If the ball is thrown upwards at an angle, it travels in a parabola. In addition, a ball bouncing on a hard surface that proceeds in a series of diminishing parabolas is caused by the energy lost on each bounce. Alternately, a cartoon character also bounces in much the same way as a ball, which is shown in Fig. 2.

According to [11], the bouncing ball simulation path has the similarity of the hop and jump of the character motion. Hence, this is to say that, the concept of bouncing ball simulation is critical to the animator as a reference for doing any kind of animation. To quote, [12] also stated that the movement is based on what happens in nature, however, simplified and exaggerated if necessary for dramatic effect. At the very least, the motion of an object should create based on the reality before it changed or modified to the motion we need. It means a realistic simulation of bouncing balls is essential as a guideline to the animators in animating their desired object motion.

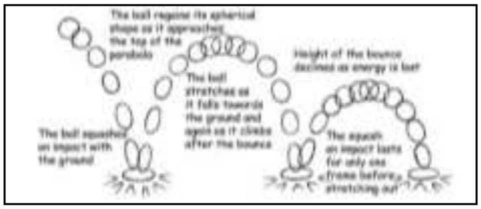

The ball regains its spherical shape as it approaches the top of the parabola
Height of the bounce declines as energy is lost
The ball stretches as it falls towards the ground and again as it climbs after the bounce
The ball squashes on impact with the ground
The squash on impact lasts for only one frame before stretching out

Fig. 1. The Concept of Bouncing Ball by [10].

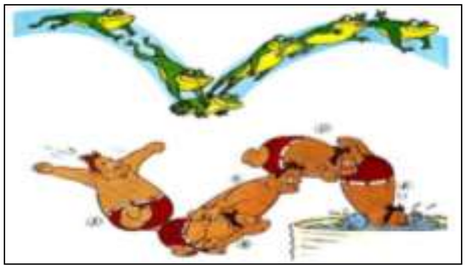


Fig. 2. Bouncing Ball and Character Motion Path by [11].

III. Analysis Physics Motion of Bouncing Simulation

Based on the review and information collected of physical measurement for bouncing balls simulation, it can be broken down into five distinct stages to analyze the details of physics calculation, to apply the accurate algorithm of bouncing simulation in animation.

In the first stage, when a ball is released from rest, it falls vertically downward under the influence of gravity (g). The velocity (V) points downward. The acceleration (a) also points downward as shown in Fig. 3. When the object is in freefall, the magnitude of (a) is equal to (g), in the absence of air resistance. The acceleration due to gravity is g = 9.8 m/s² on earth.

In the second stage, the ball begins to make contact with the surface. The ball has slowed down. The velocity (V) is still pointing downward. However, the ball has deformed sufficiently such that the acceleration (a) is now pointing upward. This means that the ball has deformed enough until it is pushing against the surface with a force greater than its weight. As a result, the acceleration (a) is pointing upward as shown in Fig. 4.

In the third stage, the ball has reached its maximum deformation when it barely touches the floor surface. As a result, the acceleration (a) is still pointing upward, and the velocity (V) is zero. The ball is preparing to re-bounce as shown in Fig. 5.

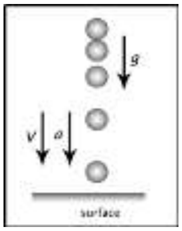


Fig. 3. The Ball in Vertically Downward Freefall.

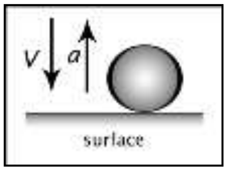


Fig. 4. The Ball Begins to Make Contact with the Surface.

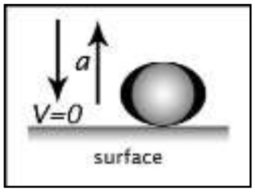


Fig. 5. The Ball Touches Floor Surface and Is Preparing to Re-Bounce.

In the fourth stage, the ball's velocity (V) is increasing and pointing upward since the ball is now in the rebounding stage. As a result, the ball is less deformed than the previous stage, but is still deformed enough until it pushed against the surface with a force greater than its weight. This means that the acceleration (a) is still pointing upward as shown in Fig. 6.

In the last stage, the ball has fully rebounded and has lifted off from the surface. The velocity (V) is still pointing upward and the acceleration (a) is pointing downward since the only force acting on the ball in this stage is gravity as shown in Fig. 7.

As shown in the above, a force would act between the ball and the floor. It changes the ball's velocity over some fairly small, but non-zero period. During this time, the ball would deform due to the force. The more rigid the ball material, the less the ball would deform, and the faster this collision would occur. In the limiting case, the ball is infinitely rigid, and cannot deform at all. To sum up, the collision is considered as occurring instantaneously in rigid body dynamics. It is the reason that the collision of the ball does not change the ball's shape in real motion. Thus, the deformation of the ball will be neglected in the factors of physics measurement.

From the above analysis, a free-fall motion under the influence of gravity can be described by the following basic motion equations:

Force (F) = Mass (m) x Acceleration (a) (1)

Velocity (v) = Displacement (d) /Time (t) (2)

Acceleration (a) = Velocity (v) / Time (t) (3)

The arrangement from the above basic equations can form four equations of linear motion which are important in solving the problems of linear motion,

$V_1 = V_0 + gt$ (4)

$d = ½ (V_0 + V_1) t$ (5)

$d = V_0t + ½gt^2$ (6)

$V_1^2 = V_0^2 + 2gd$ (7)

, where g=the constant acceleration of gravity, V_0=initial velocity, V_1=final velocity, t=time and d=displacement.

From the above analysis, a realistic formula of bouncing ball's motion need to measure in terms of the force (F), gravity (g), distance=displacement (d), time (t), acceleration (a), mass (m), and speed=velocity (v). All the physical quantity is dealing with the formula in physics law of motion. Hence, the basic physics' motion equation is very important to measure the movement of the bouncing ball.

Furthermore, physics analysis for the dynamics of the bouncing ball model was analyzed according to the graph shown in Fig. 8.

Fig. 8 shows the dynamics of squash ball bounces on the cement floors. The graph consists of the x-axis and y-axis which correspond to the variables of height, h (cm), and time, t (s) of the ball bounce. The graph represents that a ball dropped from an initial height, h_0. When it bounces, it loses energy so that the next bounce height, h_1, is smaller. It also happens in the bounce height, h_2. The model used here implied that the fraction of energy lost on each bounce. The bounce decreases by a constant amount until the ball comes to rest. Hence, the energy loss can be expressed in terms of the coefficient of restitution, COR, defined in the case of a rigid surface, is shown below:

COR = vi/vf_0 =$\sqrt{h_1/h_0}$, (8)

where, $h_0$is the initial height and $h_1$is the rebound height, $vf_0$is the initial speed of the ball and *vi* is the rebound speed.

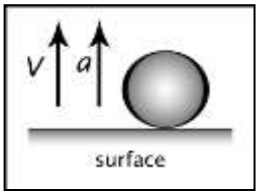


Fig. 6. The Ball is in the Rebounding Stage.

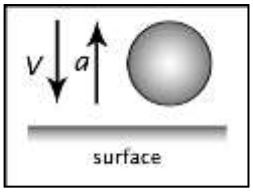


Fig. 7. The Ball Fully Rebounded and Lifted Off from the Surface.

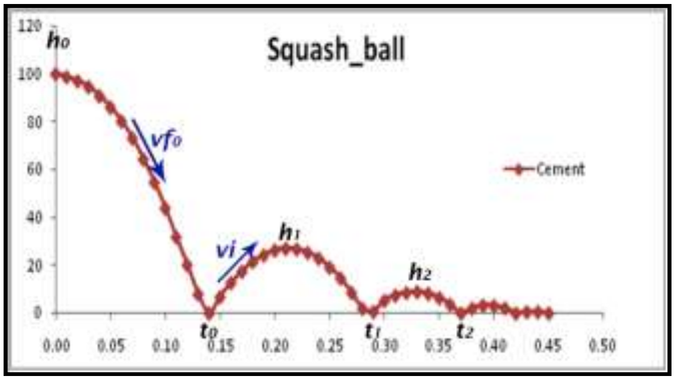


Fig. 8. The Dynamics of Bouncing Ball Model.

These parameters are important to measure the collision of the ball when it bounces and hits the floor surface. The formula of the coefficient of restitution is essential to measure the energy loss and how the ball bounces in the second height, h_1.

Besides, the motion in which a body is thrown or projected is called projectile motion. To calculate an accurate motion path of the bouncing ball simulation, projectile motion takes part in measuring the motion of the ball thrown. According to the projectile motion in Fig. 9, the physics analysis of a particle in free fall can be described by the equation from the projectile without air resistance,

$$\text{Vi} = -(\text{COR})(Vf_0) \tag{9}$$

$$h_0 = Vi^2 / 2\text{g} \tag{10}$$

$$\text{Vf} = \sqrt{2gh_0} \tag{11}$$

$$\text{t} = \sqrt{2h_0/g} \tag{12}$$

$$\text{T} = 2(\sqrt{2h_0/g}\,) - 0.01 \tag{13}$$

$$\text{h} = h_0 - (0.5\text{g}t^2) \tag{14}$$

$$h_0 - (0.5 * \text{g} * (T - t)^2) \tag{15}$$

where, vi = velocity reach top, Vf_0=initial velocity reach top, h_0 =initial height, g=gravity, Vf =velocity reach top, t=time, T=total time, COR=coefficient of restitution.

The bouncing ball simulation addresses the needs of the projectile equation. Hence, the concept of the projectile equation is consistently as one of the parameters to formulate a realistic bouncing ball simulation.

In a nutshell, the summary of physics motion measurement for dynamic bouncing ball are presented in Table I.

From the above analysis, the basic physics motions, coefficient of restitution, and projectile equations are essential in measuring the dynamics of the bouncing ball's simulation. The algorithm carries on to the next phase in exploring the formula with the real-time bouncing simulation before applicable to the animation.

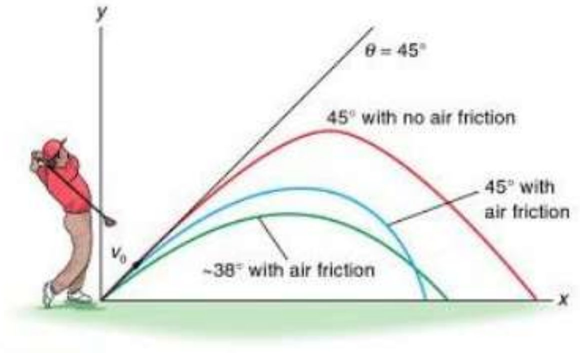


Fig. 9. The Effects of Air Resistance on Projectiles by [13].

TABLE I. SUMMARY OF PHYSICS MOTION MEASUREMENT FOR DYNAMIC BOUNCING BALL

Dynamic of Bouncing Ball	**Physics Motion**	**Formula Equations**
Energy Lost	Coefficient of Restitution	COR=$\sqrt{h_1/h_0}$
Gravity	Acceleration due Gravity on Earth	g = 9.8 m/s²
Velocity Reached Top	Projectile's Motion	Vi = -(COR)($\mathrm{Vf_0}$)
Initial height	Projectile's Motion	$\mathrm{h_0}$= $\mathrm{Vi^2}$/ 2g
Velocity Reached Down	Projectile's Motion	Vf = $\sqrt{\mathrm{2gh_0}}$
Time of ball reached top at each bounce	Projectile's Motion	t = $\sqrt{\mathrm{2h_0/g}}$
Time of ball reached down at each bounce	Projectile's Motion	t = $\sqrt{\mathrm{2h_0/g}}$
Total time reached of each half bounced	Projectile's Motion	T = 2($\sqrt{\mathrm{2h_0/g}}$) - 0.01
Height per Time	Projectile's Motion	h = $\mathrm{h_0}$-(0.5g$\mathrm{t^2}$) / $\mathrm{h_0}$ - (0.5*g*$\mathrm{(T-t)^2}$)

IV. PHYSICS MOTION AND REAL TIME DYNAMIC BOUNCING BALL

In this phase, analysis and experiments are conducted on the physic formulas to define the algorithm to be considered and applied to simulate accurate bouncing animation. The summary of physics motion measurement for a dynamic bouncing ball in Table I was used as a guideline to obtain the bouncing ball's physic motion data. In the meantime, the physics factors of the ball's properties were considered in the parameters when doing the calculation. On the other hand, the real-time bouncing ball data will be gained based on the experiment, in which a squash ball will be used and dropped from the height of 1 meter to the hard surface floor, to determine the time and height of the ball will bounce. The experiments will be recorded via a video recorder, and a stopwatch will be used to time the number of seconds between the bounces of the ball with the height of the ball bounces. The height and times are recorded in the data chart in Microsoft Excel. The experiment will be repeated for a total of 10 trials for each ball to get accurate data, and the results would be shown in the graph. The progress measurement for getting real-time dynamic bouncing ball data is presented in Fig. 10.

As shown in Fig. 10, the videos recorded were rendered to frames by using Adobe Premium Pro. The frame rate of every video was rendered every 30 frames per second (30 fps). Every frame was calculated in the Autodesk AutoCAD to have an accurate measurement. All the data was then transferred to Microsoft Excel and the real motion of bouncing balls was analysed. For a real-time bouncing ball, a dynamic hysteresis curve was presented to show how energy is lost during and after the collision.

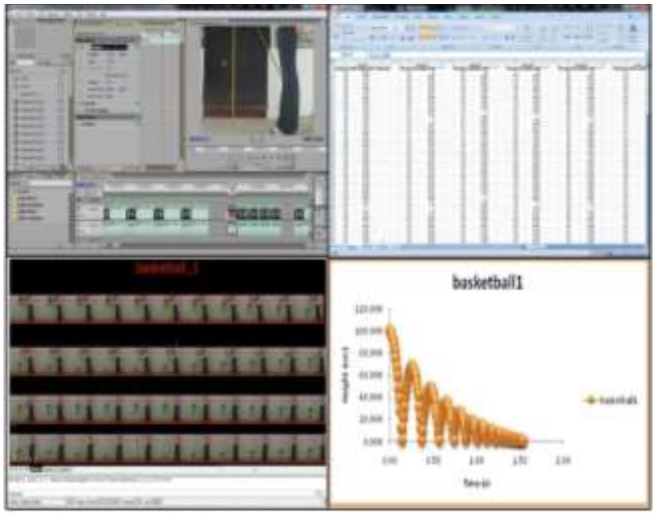
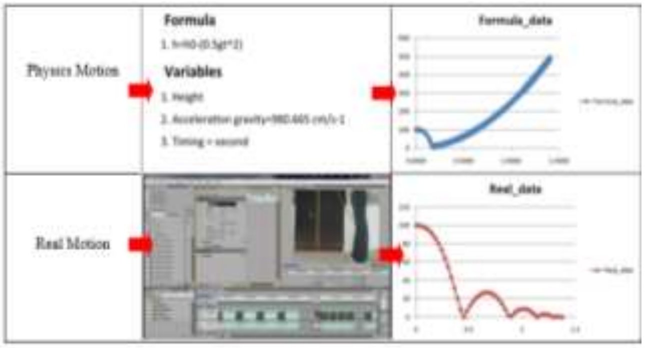


Fig. 10. Screen shot of Progress Measurement for Real Time Dynamic Bouncing Ball.



Formula

1. h=h0-(0.5gt^2)

Variables

1. Height
2. Acceleration gravity=980.665 cm/s-1
3. Timing = second

Fig. 11. Comparison between Both Physics and Real Time Data.

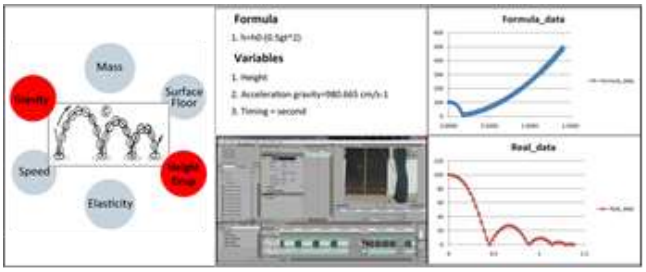
With the comparison between both physics and real-time data in Fig. 11, the accurate formulas and physics factors of the ball's properties will consider in the parameters and apply in animation.

First, the height of dynamic bouncing ball simulation is measured by using the formula of projectile motion as shown in Fig. 12. The formulas (14) are shown below,

Height per Time, $h = h_0\text{-}(0.5gt^2)$ (16)

where, h = height, t = time and g = gravity.

The variables of the object's height, gravity acceleration, and time consider in this parameter. The result shows in the blue-colored graph in Fig. 12. The comparison of physics formula, the blue-colored graph has been done with real motion data, the red-colored graph in Fig. 12. The comparison shows that this physics parameter is not sufficient to simulate the realistic bouncing ball. Based on the graph, noticed that the height for the first ball bounce is correct but it is wrong starting from the second bounce. This parameter is just for measuring the correct height for the first bounce. Hence, the parameters of measuring the second bounce's height are needed.



Formula

1. h=h0-(0.5gt^2)

Variables

1. Height
2. Acceleration gravity=980.665 cm/s-1
3. Timing = second

Fig. 12. Measurement on the Height of Dynamic Bouncing Ball.

Next, two physics formulas of projectile motion were added to measure bounce's height of the bouncing ball. The formulas used are:

Height, $h = h_0 - (0.5gt^2)$ (17)

Time reached of each half bounced, $T = 2(\sqrt{2h_0/g})\text{- } 0.01$ (18)

Height per Time, $h = h_0 - (0.5*g*(T\text{-}t)^2)$ (19)

where, h = height, t = time and g = gravity, h_0 = initial height, T = total time.

The added formulas are shown in the red color in Fig. 13. The measurements in terms of the bounce's timing and height are separated. The timing is calculated when the highest point of the ball is achieved, in which the halfway between the two bounces. Thus, the accuracy of every ball bounce's height can be found. The comparison between physics motion in the blue-coloured graph and real motion data in the red-coloured graph is presented in Fig. 13. The comparison shows that this physics parameter is less successful to simulate the realistic bouncing ball. It does not consider the fraction of energy lost on each bounce. As a result, the height is the same in every ball bounce. Therefore, the coefficient bounce height is tested and added in the formula for the next experiment, which is shown in Fig. 14.

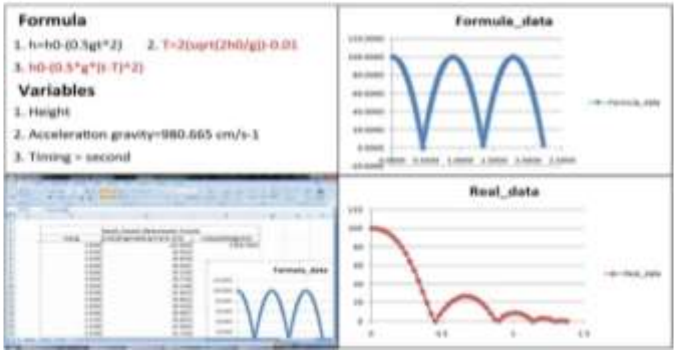
The formula of coefficient bounce height is added to measure the fraction of energy lost on each bounce. The formulas used are shown below:

Height, $h = h_0 - (0.5gt^2)$ (20)

Time reached of each half bounced, $T = 2(\sqrt{2h_0/g}) - 0.01$ (21)

Height per Time, $h = (h_0/h_1) - (0.5*g*(T\text{-}t)^{\,2})$ (22)

where, h = height, t = time and g = gravity, h0 = initial height, T = total time.



Formula

1. h=h0-(0.5gt^2) 2. T=2(sqrt(2h0/g))-0.01

3. h0-(0.5*g*(t-T)^2)

Variables

1. Height
2. Acceleration gravity=980.665 cm/s-1
3. Timing = second

Fig. 13. Measurement on the Timing and Height of Dynamic Bouncing Ball.

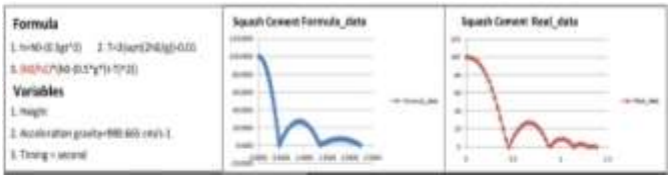


Fig. 14. Measurement on the Timing, Height and Coefficient Bounce Height of Dynamic Bouncing Ball.

The added formulas are shown in red color. The result is shown in the blue-coloured graph in Fig. 14. The comparison has been done between physics motion and real motion data in Fig. 14. From the comparison, the result shows that the graph of physics formula has a similar path compared to the real motion. But, it is still unsuccessful to measure the dynamics of the bouncing ball. It is because of the inaccuracy of the measurement on the bounce height in terms of the energy lost. Hence, the energy loss of dynamic bouncing ball is measured in the next experiment, by using the formula of the coefficient of restitution. The formulas are:

Height, $h = h_0 - (0.5gt^2)$ (23)

Time reached of each half bounced, $T = 2(\sqrt{2h_0/g}) - 0.01$ (24)

Height per Time, $h = h_0 - (0.5*g*(T-t)^2)$ (25)

Velocity reached top, $Vi = -(COR)(Vf_0)$ (26)

Initial height, $h_0 = Vi^2/2g$ (27)

Velocity reached down, $Vf = \sqrt{2gh_0}$ (28)

Coefficient of Restitution, $COR = \sqrt{h_1/h_0}$ (29)

where, h = height, t = time and g = gravity, h_0 = initial height, h_1 = second bounce height, T = total time, vi = velocity reach top, Vf_0 =initial velocity reach top, Vf =velocity reach top, and COR=coefficient of restitution.

Fig. 15 shows the measurement on the timing, height, and coefficient of restitution of dynamic bouncing ball. The formulas added are highlighted in red colour. This measurement has added the calculation on coefficient of restitution for energy loss. To calculate the coefficient of restitution for energy loss, the other formulas including velocity and initial height formula need to be added in to find the solution for it. Thus, these parameters have considered in the variables of the object's height, acceleration gravity, time and energy loss of bouncing ball. These elements which are important and essential in the real motion are highlighted in Fig. 15. The comparison between physics motion and real motion data indicated that these physics parameters can simulate the realistic bouncing ball. From the graph presented in Fig. 16, results proved that the formulas were validated to measure the bouncing ball's motion. Both motion paths achieved approximately the same result. However, there is a little difference between both bouncing ball's paths. It is because the real motion's data is gained via the average of 10 trial experiments for a type of bouncing ball. Next, the validated formulas from this experiment will be carried on to the next level, to transform them into algorithm and parameter in python script, in which applicable to animation software, Autodesk Maya to perform accurate bouncing simulation.

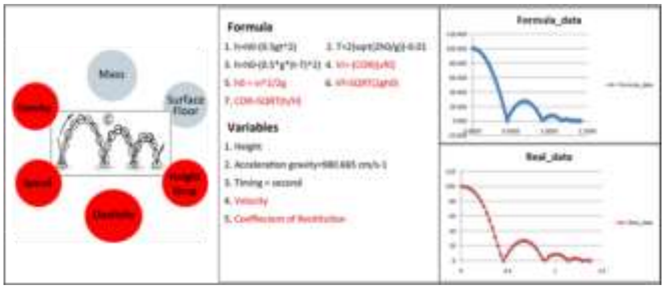


Fig. 15. Measurement on the Timing, Height and Coefficient of Restitution of Dynamic Bouncing Ball.

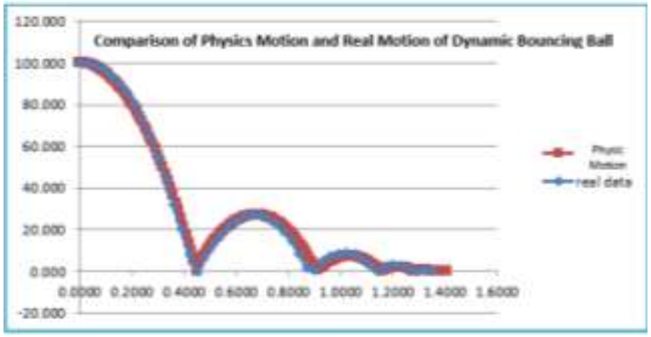


Fig. 16. Comparison of Physics Motion and Real Motion of Dynamic Bouncing Ball.

V. Results and Findings

Based on the research finding, the validated physics formulas can measure the accurate bouncing ball simulation. The summary of physics formulas and factors of ball's properties are considered as parameters will apply in animation are shown in Fig. 17.

According to the above analysis, the physics formulas are created regarding the physics factors of the object's properties. The factors of acceleration gravity, object's mass, floor surface, object's height, energy loss, speed, and time are included. All these factors play important roles in formulating a realistic dynamic of the bouncing ball. Apart from that, these parameters also involve the variables in terms of height, acceleration, timing, velocity, and coefficient of restitution for a dynamic bouncing ball. From research findings, it has been proven that the use of physics formulas would achieve realistic bouncing simulation. The physics formulas created from the above analysis are transformed into the Python scripting language as source code to drive the realistic motion of bouncing ball in Autodesk Maya, as shown in Fig. 18.

Furthermore, the comparisons are carried out between the real data via the previous experiments with the modelled bouncing ball simulation through the created system in Autodesk Maya. It is to validate the effectiveness of the applied physics formula to achieve accurate timing in bouncing ball animation. The comparisons are divided into two major elements include total time and the motion path for a bouncing ball's bounce. Every key-frame value will be calculated in the Autodesk Maya to gain accurate height and time of each modelled bouncing ball's motion. All the data are measured in Microsoft Excel and presented in the graphs. The comparison will be analysed in Table II.

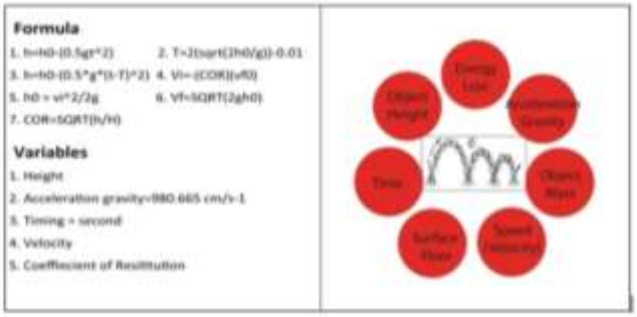


Fig. 17. Summary of Physics Formulas and Factors of Ball's Properties for Accurate Bouncing Ball Simulation.

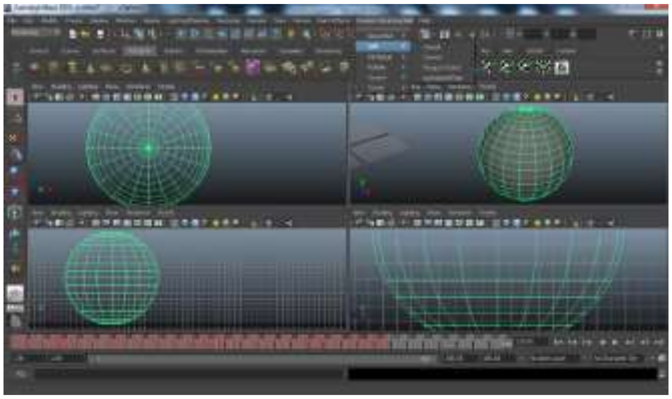
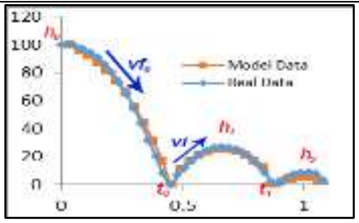


Fig. 18. Simulating Physics Based Bouncing Animation in Autodesk Maya.

TABLE II. COMPARISON BETWEEN MODELLED BOUNCING ANIMATION WITH REAL MOTION OF DYNAMIC BOUNCING BALL

Ball	**Floor**	**Bouncing Time (s)**		**Comparison Graph**
		Modelled Animation	**Real Motion**	
Squash	**Cement**	**1.10**	**1.11**	

The graphs presented in Table II show the results of the comparison between the modeled and real simulation of bouncing balls. The graph consists of the x-axis and y-axis which correspond to the variables of height (cm) and time (s) of the ball bounce. The graph represents that a ball dropped from an initial height, h_0. When it bounces, it loses energy so that the next bounce height, h_1, is smaller. It also happens in the bounce height, h_2. The model used here implied that the fraction of energy lost on each bounce. The bounce decreases by a constant amount until the ball comes to rest. Based on the Table II, a squash ball was dropped from the height, h_0 of 100cm, and reach the floor with the total time, t_0 of approximately 0.45 second. When it bounces, it loses energy, so the next bounce height, h_1 is approximately 30cm, and reach the floor with total time, t_1 of 0.85 second. It continue happens in the second bounce height, h_2 with 10 cm and finally the ball comes to rest at the time of 1.11 second for real motion of bouncing ball. On the other hand, the simulation of bouncing ball animation comes to rest at the time of 1.10 second.

From Table II, the results prove that the physically-based animation system has been validated successfully to animate the bouncing ball's motion. The modeled bouncing balls' motion paths achieved approximately the same result as in real data. However, there are small differences between both the bouncing ball's motion path and the total time of bounce. The differences occur in the time of millisecond because the real motion's data is gained via an average of 10 trial experiments for a bouncing ball. However, the motion paths of modeled balls are still approximately matched in the range of real data.

As a result, it is proven that the physics-based formula can be applied to animation as algorithm animation. Besides, realistic simulation can be achieved through created physics-based animation. The result findings also showed that the use of physics formula would create realistic animation, and achieve both physics-based realism and user-specified expressive motion.

VI. Implication and Discussion

This research studied physics algorithm formula and transformed into the Python scripting language as source code to drive accurate bouncing animation. The physically-based animation concept enables controllability on the dynamics bouncing ball in real-time without setting keyframes values in animation software. It offers controllability over a physical parameter using simple keyframe values, yet controllable animations. Besides, the application concept also allowed the animators to do the motion that would be very difficult to do by hand. Hence, it can help the animators to save more time and reduce workload while enabling faster output. The values in the keyframes tools can be changed and modified to achieve the user's desired motion. Animators are allowed to rearrange the object motion, guided by the pose specified at each keyframe. This function allows the user to further create a common model for numerous rhythmic tasks such as a falling object, walking, running, hopping, and juggling. This is because the bouncing ball simulation path has the similarity of the hop and jump of the character motion.

In addition, this research provided a physics object's factors to be considered in developing realistic bouncing ball animation, which is critical to the animator for doing any kind of animation. It is essential as a guideline to the animators in animating their desired object motion in real-time depending on their creativity. On the other hand, the algorithm can preserve the dynamics of motion and allowing animators to create motion libraries from a single input motion sequence. Once the original motion is fitted onto the model, the model can then be presented to the animator as a tool for generating a movement that meets the specifications of the given animation they are working on. The important factor of controlling real-time computer graphics is the combination of physics and animation, which helps realistically imitate real world behavior and adding to the computer graphics' degree of realism.

However, the study of the formulas for bounce characteristics is limited for common types of hard surfaces, cement floor. The balls are bounced vertically without spin on a hard, horizontal surface. Based on the review, a strategic choice of the modeler can be studying first a simpler but related problem. In this case, this could be the free fall of an object without consideration of air resistance.

VII. Conclusion

In conclusion, this study provided useful information on the development of physics-based animation. In computer animation, realistic timing is extremely important to add a life-like quality to animated objects, making their motions more interesting and able to convince the realism as well as convey the message to the audiences. However, timing of an animation is challenging to set the spatial values of the key-frame in achieving realistic simulation. It causes the burden of animation quality to rest entirely on the animator. The result findings of physics formulas in this paper provide animators a guideline that can be programmed into animation software as a library. Hence, animation such as objects falling, jumping, and bouncing animation can be produced easily without setting the keyframe value with real-time simulation.

Moreover, physically-based animation is an approach to make computer-generated virtual environments look realistic. It can be implemented in a production environment and interactive applications such as feature films, video games, or surgery simulation where the real-time simulation is required. The result of physics formulas in animation is useful to develop the games, especially the realism motion of algorithm is needed, such as shooter or bouncing game. The findings of the algorithm in this paper can be applied and transform into programming language to ensure the accurate movement of the object.

The development process of physically-based animation was explained in this paper which could be used as a reference in future research. Last but not least, future research can be expanded of the application of physics-based algorithm animation to a different type of ball’s material and bounce surface. Besides, other objects or characters such as human or animal motion, human face expression, fluid, or fur object also can be applying with the physically-based concept. A physically-based animation approach helps the animators to save more time and reduce the workload while enables faster outputs well as budget-saving. Moreover, the utilization of algorithm animation on computer animation can save the render space in computer’s memories especially when rendering a thousand animal fur or human hair in animation. This approach should be generalized to deal with other objects, so that a wider variety of controllable physics-based realistic animation can be achieved in the future.

Acknowledgment

This research was supported by the Research, Innovation & Enterprise Centre (RIEC), Universiti Malaysia Sarawak.

References

[1] I. Gris, D. A. Rivera, & D. Novick, “Animation Guidelines for Believable Embodied Conversational Agent Gestures,” In: R. Shumaker, S. Lackey, (eds) Virtual, Augmented and Mixed Reality. VAMR 2015.

[2] A. K. Kamath, N. M. Singh, R. Pasumarthy, “Dynamics and Control of Bouncing Ball,” International Conference on Intelligent Robotics and Manufacturing Automation, IRMA 2008, Vienna, Austria.

[3] A. W. Bargteil, & T. Shinar, “An introduction to physics-based animation”, SIGGRAPH '19: ACM SIGGRAPH 2019 Courses, Article No. 2, Pages 1-57, 2019.

[4] Y. Yang, J. Yang, X. Zan, J. Huang, X. Zhang, "Research of Simulation in Character Animation Based on Physics Engine", International Journal of Digital Multimedia Broadcasting, vol. 2017, Article ID 4815932, 7 pages, 2017.

[5] P. Faloutsos, “Physics-Based Animation and Control of Flexible Characters,” Master's thesis, University of Toronto, 1995.

[6] E. Promayon, P. Baconnier, & C. Puech, “Physically-Based Deformations Constrained in Displacements and Volume,” In Eurographics '96, Blackwell Publishers, pp. C155-C162, 1996.

[7] D. Terzopoulos, and K. Waters, “Techniques for Realistic Facial Modeling and Animation,” Proc. Computer Animation '91, Geneva, Switzerland, Springer-Verlag, pp. 59-74, 1987.

[8] S. Glimberg, & M. Engel, “Comparison of Ragdoll Method Physics-based Animation (Sid. 9 - 28),” Department of Computer Science, University of Copenhagen, 2007.

[9] J. Lasseter, “Principles of Traditional Animation Applied to 3D Computer Animation,” ACM Computer Graphics. Pixar, San Rafael, California, 1987.

[10] C. Webster, “Animation: The Mechanics of Motion,” London: Elsevier/Focal Press, 2005.

[11] P. Blair, “Cartoon Animation,” California: Walter T. Foster Publishing, 1994.

[12] H. Whitaker, & J. Halas, “Timing for Animation,” London: Elsevier/Focal Press, 2002.

[13] D. L. James, & D. K. Pai, “ARTDEFO: Accurate Real Time Deformable Objects,” University of British Columbia, 1999.