

## LAB #1: Motion in Two Dimensions

**BEFORE YOU COME TO LAB:**

1. Read the Orientation to Physics 103/105 lab, above.
2. Read the Introduction and Physics Background sections, below.
3. If you are unfamiliar with spread sheet programs, read Appendix A – Data Analysis with Excel
4. If you find it helpful, work the optional PreLab problem set attached at the end; You can discuss this with your AI at the beginning of the lab.

This lab is about motion, and how to describe it. The intuitive precision that a baseball player shows in moving to exactly the right location to catch a fly ball immediately after it leaves the bat is amazing. But in the technical realm it is useful to make a mathematical description. Describing motion requires measuring position and time. In Galileo's era, a measurement-based approach was in its infancy. But it is fundamental to all science.

Much of your time in this lab will be spent mastering a video-based method of measuring position and time. But don't lose sight of what is really important – characterizing motion itself. In what sense does motion sustain itself, and how do external agents affect it? Can two-dimensional motion be analyzed in terms of two separate descriptions, each involving only one dimension? Newton's simple equation,  $F = m a$  (with  $F$  and  $a$  being vectors), underlies a formal answer to such questions. But formalism has to be related to measurement and intuition. That is what this lab is really about.

## A. Introduction

In this lab you will use a video camera to make videos of moving objects, and a computer to analyze the motion. It is important that every member of your group learn how to use the computer and software. Be sure that you all take turns and that, when you are the operator, your partners understand what you are doing.

The programs that you will use can be activated by clicking on icons on your computer's **Desktop**. These include:

- **VideoPoint Capture:** We use this to record a video.
- **VideoPoint:** the program in which you will analyze your videos, by measuring positions in the video frames, and graphing and analyzing the results.
- **Excel 2003 (with WPTools):** an extension of the common **Excel** spreadsheet program, which allows more detailed calculations and graphing. The **WP (Workshop Physics)** extensions make it easy to use **Excel's** sophisticated graphing functions.
- **Student Data:** a folder with space for you to put your files.
- **Word 2003:** which can be used for note taking, and printing camera images.

The folder **Worksheets** and **Tools** will not be needed in this lab.

### Capturing a Single Still Image

Your first task is to capture a still picture of yourself to give to your lab instructor to help him or her remember your name and face.

1. Open **VideoPoint Capture**. A window pops up labeled **Choose Capture File**. The default file name is **capture**. Simply click **Save** to go to the next step.
2. A window labeled **Preview** appears, which displays whatever your camera is viewing at the moment. Wave a hand in front of the camera. The focus and iris settings of the lens should not have to be changed. Notify your AI if you believe that these settings are incorrect.
3. Position yourselves (re-aiming the camera if necessary) so that your faces fill the screen. Then reach over, hold down the **ALT** key, and hit the **Print Screen** button near the right end of the top row of keys on your keyboard. (This actually makes a copy of the screen, rather than printing anything.)
4. Now, open **Word 2003**. Click on **Edit** → **Paste** (upper left corner) to insert the image that was copied using the **Print Screen** command. (Please do NOT

change any of the computer or display settings in an attempt to make the image larger. Such changes may interfere with your analysis program.) Type your names below the image.

5. Click on **File** → **Print** icon on the top bar of **Word** to print your picture on one of the two Dell printers in the center of the Lab room. Each member of your team should include this picture in your Lab notebook. (Extra copies to take home, or to send your grandmother, are allowed.)

## **Stop and Think – What is a digital image?**

Before trying to capture a video in the computer, let's be sure that we understand what the moving image on a television screen actually consists of. It is a series of individual stationary, or still, pictures, appearing at a rate of 30 pictures, or *frames*, a second. Standard movie films consist of strips of such still pictures, recorded photographically at a slightly lower frame rate.

Digital computers, and hence our electronic imaging systems, work only in quantized units. Your camera divides the area that it sees into 307,200 pixels, arranged into a grid 640 cells wide by 480 cells high. You will see later that **VideoPoint** measures  $x$ - and  $y$ -positions with twice this resolution, dividing the  $x$ -direction into 640 integer steps and the  $y$ -direction into 480 steps. In either case, a 1-step by 1-step cell is called a *pixel*, for "picture element."

To determine the coordinates of an object in an image frame using **VideoPoint**, we just position the cursor over the object, and the system will count the pixel rows and columns to get to the location of the cursor, starting at the lower left of the image on the screen. Then, if we know how many pixels correspond to a meter long object, we can calculate real distances from the pixel counts. This is called "scaling the image." We will return to this shortly.

With real distances known from scaling the images, and with the time interval between two chosen frames also known (a multiple of 1/30 sec, say), we can easily calculate such things as velocities and accelerations of moving objects in our videos.

Note that the system reports locations rounded to the nearest integer pixel count. As a result, no position measurement can be known more accurately than within  $\pm 0.5$  pixel spacing, or to within  $\pm 1/2$  of whatever distance is equivalent to one pixel in real units (mm, meters, *etc.*).

## **B. Physics Background**

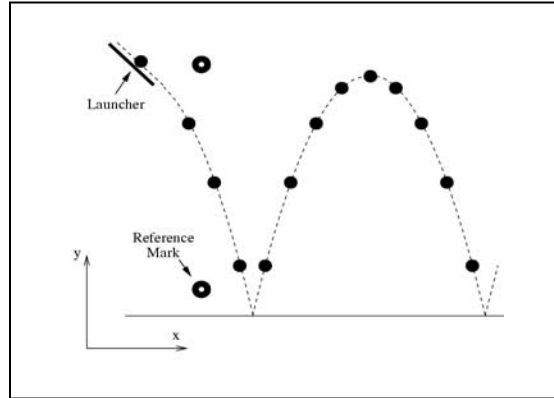
In this Lab you will analyze the trajectory of a bouncing ball. The ball will be accelerated by gravity in the  $-y$  direction, but not in the  $x$  direction. Assuming that only gravitational forces are significant, the ball's motion should obey the equations

$$x = x_0 + v_{0x} t , \quad (1)$$

$$y = y_0 + v_{0y} t - 1/2 g t^2 , \quad (2)$$

$$y = y_0 - x_0 v_{0y}/v_{0x} - g x_0^2 / 2 v_{0x}^2 + x (v_{0y}/v_{0x} + g x_0 / v_{0x}^2) - g x^2 / 2 v_{0x}^2 . \quad (3)$$

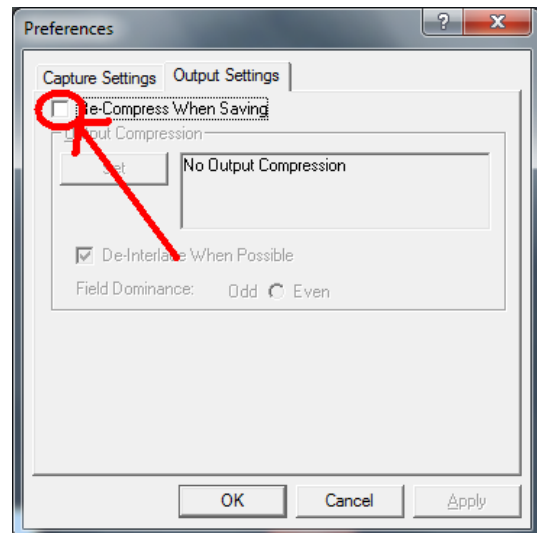
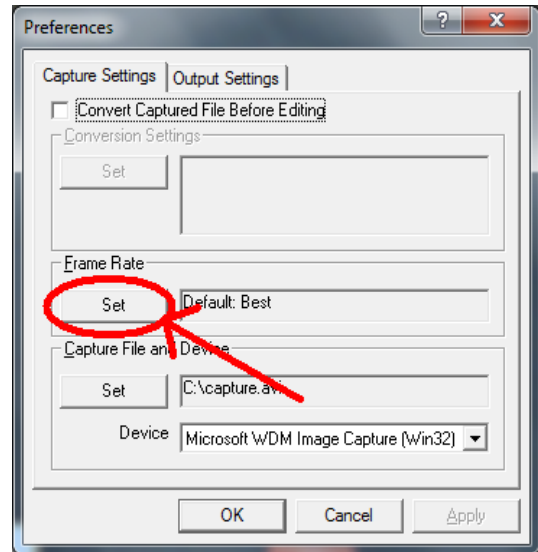
In this experiment, a golf ball rolls down a fixed rail (the “launcher”). After it leaves the launcher it falls freely, bouncing at least twice on the table top. The camera takes a series of pictures at fixed time intervals between the first and second bounces, when the ball is moving freely. By measuring the position of the ball in each picture, you can test equations (1) and (2).



### C. Acquiring the Data

Open VideoPoint Capture and use the Preview Screen to

1. Aim the camera, and check the lens focus and aperture setting.
2. Get a meter stick from the center of the room, and place it with its support bracket against the black backdrop. Verify that the launching rail is the same distance from the backdrop as is the meter stick. Remove the meter stick.
3. Make a few trial runs, adjusting the launcher as necessary for the camera to catch one full bounce.
4. Set the frame rate to 30 frames per second, using Edit → Preferences → Capture Settings. Then, in the tab Output Settings, be sure the box next to Re-Compress When Saving is NOT checked. You must do this each time you (re)open VideoPoint Capture.



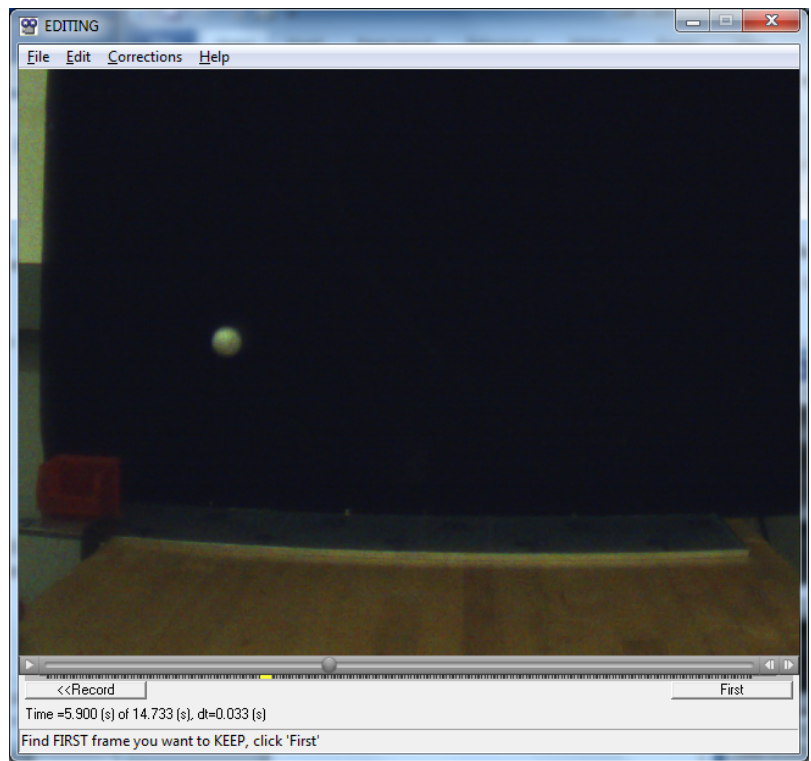
## Make an Actual Movie of the Bouncing Ball

1. Start recording by clicking on the **Record** button.
2. Start the ball rolling down the launcher, and let it complete one full bounce.
3. Hit the **Esc** button on your keyboard to stop recording. (The **Stop** button on the screen doesn't always stop the recording quickly.)
4. Note: Although you don't need to rush things, don't record too much “dead air” before and after the actual experiment – it just ties up computer memory.
5. If you need to remake your video, click on the **File** menu (upper left) and then on **New Capture** (and don't **Keep/Save** the bad video). Or, click the **<<Record** button in the lower left . Or, you can quit and restart **VideoPoint Capture**.

## Viewing Your Video

After **VideoPoint Capture** takes a few seconds to organize your video, it displays the first frame of your movie in an **EDITING** window. Note that you can:

1. Play the movie by clicking on the little triangle at the lower left. Watch the progress as indicated by the moving “progress button” below the image. (Clicking on the triangle while the movie is playing will stop it at whatever frame it has reached.)
2. Jump to any frame by manually moving the progress button with your cursor.
3. Single-step from one frame to the next by clicking on the left- and right-arrow buttons in the lower right corner.



Each of you should spend some time playing with the **EDITING** screen controls, to understand their functions and to get a “feel” for the fact that a video or movie is really only a sequence of still pictures.

## Editing Your Video

In general, your original video will contain more frames than you will want to use for analysis. Today, we want to analyze the ball's motion only between two consecutive bounces off the table.

To get rid of the unnecessary frames at the beginning and end of your video:

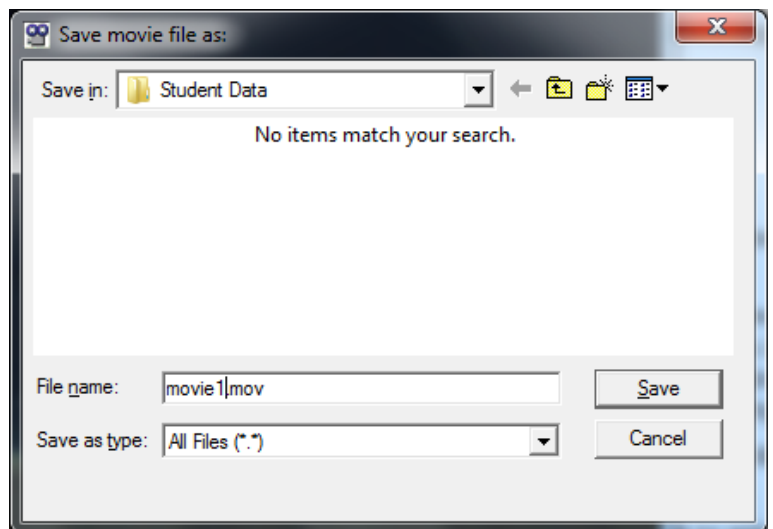
1. Use the progress and single-step buttons to choose the first frame after the ball has bounced from the table.
2. Click on the **First** button. All of the earlier frames are eliminated.
3. Move the video back to the last frame before the ball bounces again. Then click on **Last**. All of the later frames are eliminated, and your video contains just one bounce.
4. Click on **Keep** to save the file (or click on **File**, and then **New Capture** to discard the data and record a new movie).

## Naming and Saving Your VideoPoint Capture File

There is a folder called **Student Data** on your **Desktop** in which you should store all of your data and analysis files.

To save your video,

1. Click on the **Save** button on the editing screen.
2. Enter a name with a **.mov** extension in the window **Save Movie File as**.
3. Click **Save**.
4. You are done with the program **VideoPoint Capture** for now. If you **Exit** and restart it later, you will have to reinitialize the frame rate and uncheck the **Re-Compress** box.

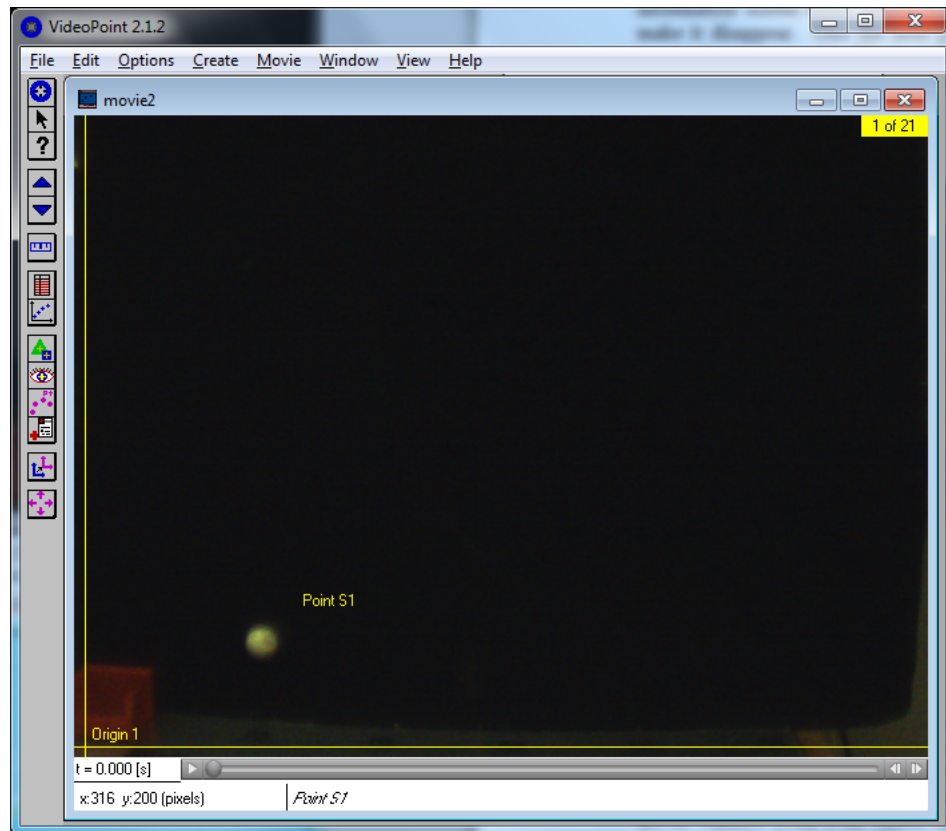


## D. Digitizing the Ball's Trajectory Using VideoPoint

Click the VideoPoint icon on your Desktop to open this program. If you see an information screen titled **About VideoPoint**, just click on the little x at the top right to make it disappear. One the next popup window, click **Open Movie**, and click on the movie that you saved in **Student Data**. In the **Number of Points** window which then appears, click **OK** to 1, since there is only one object that we want to analyze (the bouncing ball).

Your movie should now appear in a **VideoPoint** window, that shows the first frame of your edited video, with your file name appearing in its title bar. *Two other windows pop up also, **Table** and **Coordinate Systems**, which may be behind the movie window.* The movie window shows one frame at a time, whose number, and the total number of frames are given in the upper right corner.

Confirm that you can step through your movie, using the double arrows at the lower right. If the ball is not visible in all frames, or the movie includes more than a single arch, you should go back and make a new movie.



### Digitizing Coordinates of the Bouncing Ball

The movie window is where you will "pick off" the coordinates of the golf ball's position in each frame. Note the yellow coordinate axes on the screen, and note that the cursor changes to a cross-hair pointing device when you move it into the image area. Finally, note that both the time of the frame and the  $x$ - and  $y$ -coordinates of the cursor are shown at the lower left of the movie window (with even values, equal to twice the pixel coordinates).

To record your positional data,

1. Go to the first frame of your edited movie.
2. Position the cursor over the image of the golf ball, and click your left mouse button.
3. Note that the movie advances automatically to the next frame. (If it doesn't, go the **Options** menu and put a check mark next to the **Auto Frame Advance** option.)
4. Click on the image of the golf ball for several more frames. Pause to turn on the **Edit → Leave/Hide Trails** option, and note that all the point that you have digitized are now shown, as well as all frames being superimposed.
5. Continue clicking on the image of the golf ball until you have recorded its position in all frames of your movie.
6. If you made a mistake in positioning your mouse in a particular frame, you may want to correct this. Go to the **Coordinate Systems** window and highlight the row labeled **Point S1**. This changes the symbol of the digitized point in the active frame to two concentric circles. *If this is not the point you wish to change, go to the frame that contains the point in question.* Then, while holding down the left button on the mouse, move the location circle to the new, corrected, position. Releasing the mouse button completes the change, and enters the corrected coordinates into the data table. *It may be easier to first move the cursor to a very “wrong” position, and then make the “right” measurement.*

The **Table** window now contains a list of your digitized points. It is prudent to save your work via **File → Save As)** to **Student Data**. **VideoPoint** will create a **.vpt** file for this.

*VideoPoint includes various options for graphing and fitting curves to data. While you are free to try these out, we recommend using Excel for this purpose. The Appendix gives advice on how VideoPoint could be used instead.*

#### **Summary of Options When Picking Off Coordinates:**

- Note that the *Movie* cursor looks like a crosshair if no coordinate has been chosen for a given frame, and like an arrow otherwise.
- Use **Edit/Leave** or **Hide Trails** to turn trails on or off.
- Use the **Coordinate System** window to **Select** a point series by highlighting it
- Move a *selected* point in a given frame, by positioning the tip of the arrow on the point and holding down the left button.

## **E. Using Excel for the Data Analysis**

Spreadsheet programs are powerful tools for data analysis, in finance and other fields as well as in science and engineering. Our version, **Excel 2003** with includes extensions generated by an academic project, **Workshop Physics (WP)**, to allow easier generation of graphs from spread sheet data.



Start Excel to obtain a new, blank worksheet. To transfer your digitized data from VideoPoint, you will “cut and paste”. Back in VideoPoint, highlight all the data in the time column, and click Edit → Copy Data. In Excel, right click on cell A2, and then click on Paste. Type a column label in cell A1. This label will later appear on your plots.

Similarly, highlight the two columns of your  $x$  and  $y$  data in the VideoPoint Table, then right click on cell B2 and click Paste. Add column labels in cells B1 and C1.

To make a graph in Excel with WPTools, you have to identify what you want plotted:

1. First, “swipe” your cursor across the data you want to plot on the horizontal axis. (For example, swipe across the time values.)
2. Then, while holding down the Ctrl key, swipe across the corresponding data you want for the vertical axis (say, the  $y$  values).
3. In the WP Standard bar click on the Scatter Plot icon to generate the plot.

A window should appear on your .xls sheet showing a plot of  $y$  vs.  $t$ . Create plots of  $x$  vs.  $t$  and  $y$  vs.  $x$  in the same manner. These plots use your column labels.

You can now address the main issue of this Lab: does the bouncing golf ball obey equations (1)-(3). To do this, you will fit curves to your graphs, and thereby determine experimental values for  $x_0$ ,  $y_0$ ,  $v_{0x}$ ,  $v_{0y}$  and  $g$ , as well as estimates of the accuracy of these measurements.

First, consider eq. (1) using your plot of  $x$  vs.  $t$ . This is a linear equation, but it’s best to try a quadratic fit to learn if the data is actually linear.

1. Click on one of the data points of your  $x$  vs.  $t$  graph, to select and highlight the points plotted.
2. Click the Polynomial Fit icon in the WP Standard bar and choose Order 2 to create the fitted function. We recommend that you choose the order to be one higher than that you believe describes that data: thus, choose order 2 for supposedly linear data. Several boxes describing the fit should appear on your graph window. [If not, go to WPTools / Preferences and make sure that the options for displaying fit equations and statistics on the plot are both checked. If not, check them, and redo the plot.]
3. The parameters of the fit are

$a_0$  = the constant term in the quadratic equation

$a_1$  = the multiplier for  $t$  in the equation

$a_2$  = the multiplier for  $t^2$ .

- Note the value for  $\sigma$  (lower-case Greek letter sigma). This is the computer's estimate for how far a typical data point deviates from the fitted curve.

*Does the value of  $\sigma$  seem reasonable? You should comment on the magnitude of the typical deviations, and on possible causes, in your notebook. If  $\sigma$  is too large, you should redigitize your images, or take a new movie.*

- Note also the values for  $SE(\mathbf{a}_0)$ ,  $SE(\mathbf{a}_1)$ , and  $SE(\mathbf{a}_2)$ . These are values of what are called the "standard errors" in the values which the computer calculated for  $\mathbf{a}_0$ ,  $\mathbf{a}_1$ , and  $\mathbf{a}_2$ . These are the computer's estimates for how precisely your data determine the fit parameters, given the "jitter" of the data about the curve. *A discussion of the algorithm used to make these standard error estimate is given in Appendix D of the Lab Manual.*
- You can consider Excel's value for  $\mathbf{a}_1$  and  $SE(\mathbf{a}_1)$  to be your physics result for the value of  $v_{0x}$  and the estimated error in that value (still in units of pixels/sec, of course, since you haven't yet converted pixels to meters.) Record your result for  $v_{0x}$  in your lab book, and calculate the relative precision,  $SE(\mathbf{a}_1) / \mathbf{a}_1$ .

Make similar analyses of your plots of  $y$  vs.  $t$  and  $y$  vs.  $x$ , using polynomial fits of order 3 since the physics expectation is that these data should be quadratic functions. Of particular interest are the values (and error estimates) of parameters  $\mathbf{a}_2$  from these fits, since they are proportional to the acceleration  $g$  of gravity, according to equations (2) and (3).

### A Common Problem

Did your graph of  $x$  vs.  $t$  have some curvature, rather than looking like a straight line as implied by eq. (1)? A common reason for this is that the plane of the trajectory of the golf ball was not perpendicular to the optical axis of the camera.

If the horizontal angle between the optical axis and the perpendicular to the plane of the trajectory is  $\theta_x$  in radians, then the coefficient  $a_2$  in your fit to  $x$  vs.  $t$  is approximately

$$a_2 \approx \frac{v_{0x}^2 \theta_x}{D},$$

where  $D$  is the distance from the camera (focal plane) to the plane of the trajectory. A consequence of this is that your measurement of  $g$  via the  $y$  vs.  $t$  plot has a systematic error roughly given by

$$\frac{\Delta g}{g} \approx \theta_x \left( 1 - \frac{2v_{0x}v_{0y}}{gD} \right).$$

This effect could well be a major source of error in your measurement of  $g$ , so you may wish to make new movies of the bouncing ball until you get one with a "small" value of  $\theta_x$ .

### Interlude

Before we convert pixels to meters and get our final physical result for the value of  $g$  and its uncertainty, let's review some of the concepts we have used.

1. Things are accelerated by gravity in the vertical direction, but not horizontally.
2. Constant acceleration leads to a quadratic variation in position.
3. An electronic camera can measure positions in pixels, which have later to be converted to meters. The integer-pixel position measurement leads to an uncertainty of  $\pm 0.5$  pixel in our knowledge of the true position (and VideoPoint multiplies all pixel values by 2).
4. Every individual measurement is inexact, and we need to understand what a typical uncertainty in a measured quantity might be.
5. Our result for the value of  $g$  (in pixels/sec<sup>2</sup>) is inexact, because the measurements are inexact.

Our focus on the deviations of measured points from the fitted line assumes that the fit itself is our best measure of the true function, and that the deviations are due to *random jitter* in the individual measurements. However, other types of errors, called *systematic errors*, can affect all of your measurements in a similar manner. An example occurs when you convert pixels to meters. The conversion factor clearly affects all of your points at once.

## F. Scaling the Pixels

We have now to determine how many pixels in our picture correspond to one meter in the real world. We will do this by measuring how long a meter stick was in our video, as measured in pixels. The resulting value for our conversion factor (pixels per meter, or the inverse, meters per pixel) will allow us to convert any measurement from pixels to meters (or pixels/sec to meters/sec, *etc.*).

Our conversion factor will have some error, which we can estimate based on the pixel quantization error, plus a “guesstimate” as to how reproducibly we can position our cursor on the ends of the meter stick. Once we know the conversion factor, and its estimated error, we can use it to convert all of our measured  $x$ - and  $y$ -positions to meters from the origin, rather than pixels. But the conversions will all involve the factor, which has some error associated. This is a systematic error, since it affects all measurements equally.

If VideoPoint Capture is not still running, restart it (and remember to initialize the frame rate and uncheck the **Re-Compress** box). Take a short movie with the meter stick vertical in the center of the frame, and another with it horizontal. Remember to place the meter stick as far from the backdrop as is the launcher rail.

Use VideoPoint to analyze these movies. Move the cursor manually to either end of the meter stick (or the 10- and 90-cm positions), and note the pixel coordinates at the two position, from the live display at the lower left corner of the movie screen. Calculate the

distance between the two points, in pixels. Using the distance in pixels, calculate your movie's scale factor,  $F_x$  (or  $F_y$ ), in units of pixels per meter. Use this to convert the relevant fit values from your three plots into meters (or m/s, or m/s<sup>2</sup>, etc.) to deduce the acceleration of the golf ball in m/sec<sup>2</sup> by two methods. Note the result in your lab book.

*Are your answers close to the expected value of  $g$ ? What effects might systematically bias your answer away from 9.8 m/s<sup>2</sup>?*

Now let's consider what is the accuracy with which your measurements determine a value for  $g$ . If the uncertainty in  $g$  is called  $\Delta g$ , then the fractional, or relative, uncertainty in  $g$  is defined as  $\Delta g/g$ .

When you estimate the uncertainty in your determination of  $g$ , you should use both the computer's estimate of the effects of random error,  $SE(a_2)$ , and your estimate of the uncertainty in the conversion factors  $F_x$  and  $F_y$  from pixels to meters. That is, for the measurement using the  $y$  vs.  $t$  data,

$$\frac{\Delta g}{g} = \sqrt{\left(\frac{SE(a_2)}{a_2}\right)^2 + \left(\frac{\Delta F_y}{F_y}\right)^2},$$

where  $F_y$  is your  $y$  scale factor. It is up to you to estimate a value for  $\Delta F_x$  and  $\Delta F_y$ , perhaps by repeated trials. Taking responsibility for estimating the accuracy with which you make a measurement is not easy, but you have to do it. Take time to think it through!

Similarly, the estimate for the relative uncertainty on  $v_{0x}$  as determined from the  $x$  vs.  $t$  plot is

$$\frac{\Delta v_{0x}}{v_{0x}} = \sqrt{\left(\frac{SE(a_1)}{a_1}\right)^2 + \left(\frac{\Delta F_x}{F_x}\right)^2},$$

and the relative uncertainty on  $g$  as determined from the plot of  $y$  vs.  $x$  is

$$\frac{\Delta g}{g} = \sqrt{\left(\frac{SE(a_2)}{a_2}\right)^2 + 4\left(\frac{\Delta v_{0x}}{v_{0x}}\right)^2}.$$

*Some discussion of the calculus of propagation of errors is given in Appendix B.8 of the Lab Manual.*

### **Record your final results for $g$ and for $\Delta g$ .**

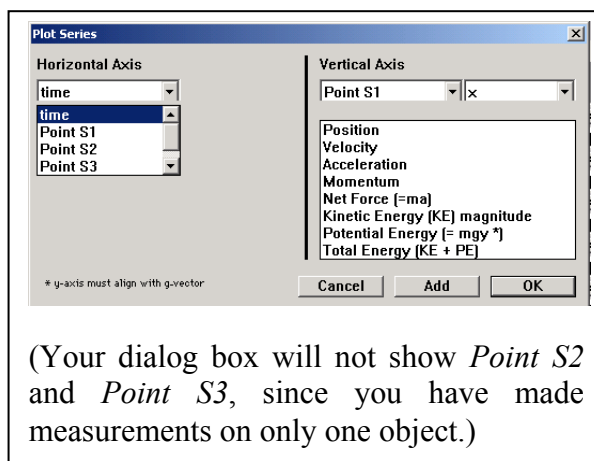
Note any comments you might have on the reliability of your conclusions, and on a comparison of your results with the accepted value for  $g$ , namely 9.80 m/sec<sup>2</sup>.

## Appendix: Using VideoPoint for Data Analysis

### Graphing Your Data

To use VideoPoint to make graphs, first click on the View → New Graph menu item. For each graph, you must tell the system which variable you want plotted on the  $x$ -axis, and which on the  $y$ -axis. Normally, you will want either time or  $x$ -position to run horizontally, and  $x$ - or  $y$ -position, velocity, or acceleration to run vertically.

Let's make and print out some graphs. Start with an XY graph. Choose Point S1 /  $x$  / Position for the Horizontal Axis, and S1 /  $y$  / Position for the vertical axis. Note that, if you go back to the *Movie* window and advance through the frames, a circle moves on the graph to indicate which of the points plotted corresponds to the movie frame currently visible.



To print a graph, first make its screen active by clicking anywhere on it. (The title bar of the graph will be colored blue after your click.) Then click on the File → Print menu bar option, and click on Print and OK when you are asked to confirm your request. If you maximize the size of the graph window before printing, your printed graph will be larger.

You can also make a printout of any one of the frames of your video, simply by printing when the movie window is selected.

If you need a printout of your data table, first be sure to first maximize the table window by clicking on the middle one of the three boxes at the right end of the **Table** title bar. Then issue the print commands. When you want to make the table small again, just click on the middle box, as before. (If your table is very long, you may have to print more than one time, scrolling vertically to change which data lines are presented each time.)

### Physics Discussion

Print out and think about the following graphs. Each of you should have your own copy of the group's printouts, and **note your own conclusions on your own plots.**

1. Do the plots of  $x$ -position, velocity, and acceleration all confirm that the golf ball is subject to no horizontal force?
2. Does a plot of  $y$ -position vs. time look like you expected? (What is that?)
3. What does a plot of  $y$ -velocity vs. time look like? What does that show?

#### 4. What about a plot of $y$ -acceleration vs. time?

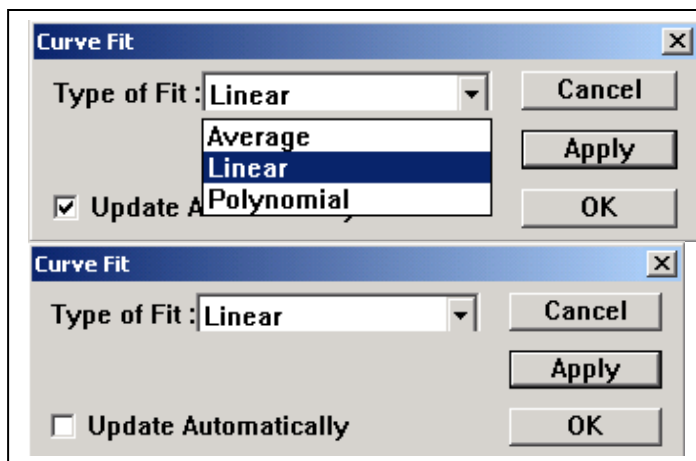
You should know that we expect to see a constant, negative, acceleration of the golf ball in the  $y$ -direction. Numerically, we expect that the acceleration of the golf ball will be  $g$ , which is approximately equal to  $9.80 \text{ m/sec}^2$ . But we have yet to scale our movie, in order to find the conversion factor between pixels and meters.

Before we scale the movie, let's look at our results in terms of pixels, and take a look at how accurate those results seem to be.

### Final Analysis – Fitting Your Data; Looking at Data "Jitter"

Now for some more quantitative analysis. First, let's fit a quadratic curve to  $y$  as a function of time. The computer can show us the quadratic function that "comes closest" to passing through all of the data points on our  $y$  vs.  $t$  plot.

To do this, go to your graph of  $y$ -position vs. time, and then click on the red F (for Fit) button near the upper right hand corner. A



dialog box will appear. Select the Polynomial option, and then choose Order 2, and click on Apply. VideoPoint's best-fit line should appear on the graph.

At the top of the graph is the algebraic function resulting from the plotted fit. (Here,  $x$  refers to whatever is the horizontal variable on the graph, not to our  $x$  coordinate. Also, you should ignore the computer's  $R^2$  (R-squared) parameter. It is simply one of several statistical measure of the goodness of the fit, and VideoPoint often calculates it incorrectly.)

The number multiplying the squared term in the fit corresponds to the " $1/2 g$ " factor in equation (2). If we multiply our number by 2, that gives our experimental value for  $g$ , in units of pixels/sec<sup>2</sup>.

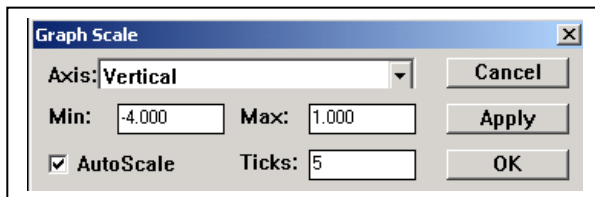
Before moving on to scale the movie, and get  $g$  in m/sec<sup>2</sup>, let's look at how closely our data follows the computer's "best fit" curve.

### Zooming in VideoPoint; Looking at Data "Jitter"

Maximize your plot of  $y$  vs.  $t$ . Probably all of the points lie close to the curve, but how close? "Magnify" a region of your VideoPoint plot, by "zooming in" on it with the following steps:

5. While holding down the **Ctrl** (Control) key of your keyboard, hold down on the left mouse button and use the cursor to draw a rectangular box around a region of interest containing one of your data points. The graph then shows only that region, in a magnified view.
6. If you want to magnify some more, do it again. (But if you magnify too much, the fit line may disappear, and you'll have to start over by returning to the unmagnified graph.)
7. When you want to return to your original unmagnified view, just hold down on the **Ctrl** key and double-click with the left button of your mouse.

Use the zooming feature to look at several points of your graph, and to judge how far the points are vertically away from the best-fit curve. **Record your results, and calculate the average of the magnitudes of the deviations that you found.** [WARNING – after a “zoom,” the scales on VideoPoint's graphs may not be quite what they seem. Often, there is an additional decimal place which is not shown on the screen's axis labels. To correct this, you can double-click on the axis, and delete the extra digit to cure the problem. For now, it is easier to use the cursor readout at the lower left corner of the graph to estimate the differences between your data point and the nearby curve.]



The average deviation which you have calculated is an estimate for how much an individual measurement varies from the “true” value indicated by the curve. It includes the random  $\pm 0.5$  pixel “quantization error,” plus any inaccuracies you introduced in picking off your points.

## Storing Your VideoPoint File

First, click on **File** → **Save As** in the VideoPoint menu bar. As before, use the up arrow and double-clicking to cause your section’s folder identification to appear in the **Save in:** box at the top of the dialog box which appears. It will be convenient to use the same name as the one you gave the video file which you saved at the end of the **VideoPoint Capture** activities. This name should be visible at the top of the movie screen, if you have forgotten it.

If you use **Windows Explorer** to look at the contents of your section’s data folder, you will see that there are actually three files associated with any movie that you have analyzed. A large file with the extension **.mov** contains the actual movie. A small file, having the same name with the extension **.mov.#res** added comes along with the movie file. The **VideoPoint** file **.vpt** is also a small one. It has whatever name you gave it, but can be used only if the movie file is also available on your computer.

## PRELAB Problems for Lab#1: Motion in Two Dimensions

1. Assume that in a digital image there are 600 lines and 800 columns (typical for some computer display screens). Each point in the image lying at the intersection of a line and a column is called a picture element, or “pixel.” Each pixel has its own brightness and color. The intensity of each of the three primary colors at each pixel may be described by an integer number, lying between 0 and 255. Each number takes 8 digital bits, or 1 “byte,” to define.

- (a) How many pixels are there in the image?
- (b) How many bytes of computer memory are required to record the image?
- (c) If a video or movie consists of a series of 30 such images every second, how many bytes of information must be transmitted every second in order to display the movie? How many bits per second? (This is what sets the electrical engineer's design specification for a television system.)

2. Isaac Newton is said to have considered the motion of an apple falling from the tree in thinking about gravitational forces. If we look at the apple's vertical motion, starting from the instant when it is released from rest, the distance it falls increases with time according to the formula for uniform acceleration,

$$s = 1/2 g t^2 .$$

Galileo, who died the year Newton was born, studied motion under uniform acceleration, but did not have Newton's mathematics. He concluded that

*Starting from rest, distances traveled in successive equal increments of time are in the proportions 1 : 3 : 5 : 7 : .....*

- (a) Using the formula  $s = 1/2 g t^2$ , calculate the vertical distances traveled by a ball falling from rest after 0.1, 0.2, 0.3, 0.4, and 0.5 seconds. (To honor Newton's heritage, let's use English units and take  $g = 32 \text{ ft / sec}^2$ .)
- (b) Show that your results are consistent with Galileo's description.

If you are intrigued by this result, you might want to prove that Newton's result produces Galileo's rule, for any set of equal time increments. Hint: think about the quantity  $[(N + 1) \Delta T]^2 - [N \Delta T]^2$ .