

# Video Input Driven Animation (VIDA)

Meng Sun

Allan D. Jepson

Eugene Fiume

*Department of Computer Science  
University of Toronto  
Toronto, Ontario M5S 3G4  
jepson@vis.toronto.edu*

*meng@dgp.toronto.edu*

*elf@dgp.toronto.edu*

## Abstract

*There are many challenges associated with the integration of synthetic and real imagery. One particularly difficult problem is the automatic extraction of salient parameters of natural phenomena in real video footage for subsequent application to synthetic objects. Can we ensure that the hair and clothing of a synthetic actor placed in a meadow of swaying grass will move consistently with the wind that moved that grass? The video footage can be seen as a controller for the motion of synthetic features, a concept we call video input driven animation (VIDA). We propose a schema that analyzes an input video sequence, extracts parameters from the motion of objects in the video, and uses this information to drive the motion of synthetic objects. To validate the principles of VIDA, we approximate the inverse problem to harmonic oscillation, which we use to extract parameters of wind and of regular water waves. We observe the effect of wind on a tree in a video, estimate wind speed parameters from its motion, and then use this to make synthetic objects move. We also extract water elevation parameters from the observed motion of boats and apply the resulting water waves to synthetic boats.*

## 1. Introduction

The success of computer animation as an expressive medium has put increasing demands on integrating computer animation with real video or film footage. Techniques from augmented reality and inverse rendering now permit the rendering of synthetic models within video sequences. However, inserting objects that interact with or are affected by forces and real objects present in the video is a difficult open problem. How do we ensure that the hair on a synthetic actor placed in a meadow of swaying grass will move consistently with the real wind field that moved that grass?

In this case, raw video footage can be seen as a kind of controller for the motion of the synthetic hair. We call this concept *video input driven animation*, or VIDA. The schema we propose for VIDA is to analyze an input video

sequence, to extract relevant parameters of the motion of objects in the video, and to use these parameters to drive the motion of synthetic objects are introduced into the real environment. One aim of VIDA is to allow synthetic objects to be natural participants in a video. This opens up a huge and fascinating range of problems.

It is difficult to tackle these problems in a unified, comprehensive way. Rather, it would appear that different approaches will be needed for different classes of phenomena. In this paper, we shall validate VIDA by studying the problem of inverse harmonic oscillation and use it to “reverse engineer” natural phenomena such as wind or regular water waves. Two examples will be used to demonstrate the strength and applications of this methodology. The first is related to wind-object interaction. We observe the effect of the wind on a tree in a video, estimate the wind speed from the tree’s motion, then use the wind information to make computer generated objects move. The second example is related to boat-water interaction. We study the motion of boats anchored by the shore, estimate the water wave elevation that drives the boat motion, then use the water wave parameters to control synthetic floating structures.

We employ a mixture of techniques from the computer graphics, computer vision, and the hydrodynamics literature. Our results are novel to all three areas, but here we emphasize the computer vision aspect. The overall goal is to provide tools for animators to construct physically-sound and consistent animations consisting of real and synthetic objects interacting with one another. The results we present in this paper are a step toward this goal. We stress that our emphasis is less on the final rendered quality than on inverse modelling algorithms.

### 1.1. Related Work

VIDA can be viewed as a type of indirect puppetry. Several research papers have made significant contributions in this direction. Brand’s work in “Voice Puppetry” [4], introduced a method for predicting a control signal from another related signal, and applied it to voice puppetry: gen-

erating full facial animation from expressive information in an audio track. The approach involves using one signal to control another type of signal by studying their correlation. Though it has a similar flavor to VIDA, but we are exploring quite different research areas. Popović and Witkin [15] introduced an algorithm for transforming character animation sequences that preserve essential physical properties of the motion. They take the approach of motion transformation as the underlying paradigm for generating computer animations. Their work explores the forces or the spacetime optimization parameters within one object and eventually adapt them to other objects. Our work instead explores parameters for natural phenomena such as wind and water waves, which introduces an indirect level of parameter acquisition.

Increasingly, computer graphics has turned to video input for inspiration. Schödl et. al. [20] introduced a video based animation technique called *video texture* that creates a continuously varying stream of images derived from the original video stream. Our work instead analyses video for the purposes of parameter extraction. We can then use the parameters to animate objects that are significantly different from the real objects captured on video. For instance, after estimating the wind speed parameters in the scene by observing the tree movement, we can apply the wind to snow, dust, hair, clothes, etc.

Researchers in computer vision have also studied the problem of recovering physical parameters of objects and environments from video data. Bhat et al. [1] present a framework for computing the physical parameters underlying the motion of a tumbling rigid object in free flight. Their optimization framework identifies parameters such as gravity, inertia and initial velocities from video. This work has a similar flavor to VIDA, but we are exploring different research directions and applications.

Researchers in computer graphics have always been fascinated by natural phenomena. For example, Shinya and Fournier [21], Stam [23] and Shinya et al. [22] have provided solutions to the problem of computing a forward simulation of a plant’s motion in the presence of supplied wind field. While we can take advantage of this work once we have found the wind field, our job is in fact to estimate the wind field given the behavior of real objects such as trees that are moved the wind field.

The dynamics of boat motion in regular water waves has been studied mainly in naval architecture and marine engineering [2, 9, 11, 12, 13, 16, 18, 5]. To our knowledge, there is no literature in computer graphics on the depiction of boat motion under the influence of regular water waves, a visual model for which we present below. We also estimate regular water wave elevation parameters by observing boat movement, which is novel to both computer graphics and computer vision.

We now summarize the remainder of the paper. Section 2

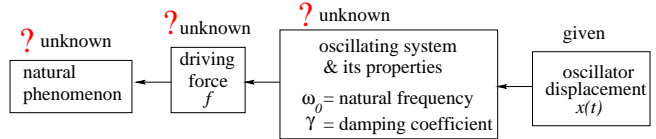


Figure 1: Estimating the driving force.

describes our solution for inverting a harmonic oscillator. Section 3 explores the basic components in extracting useful parameters by observing harmonic oscillation. Section 4 discusses the example of “reverse engineering” wind speed parameters by observing object motion due to the wind. Section 5 presents the example of estimating water wave elevation parameters by studying the motion of boats anchored at shore. Section 6 summarizes our contributions and outlines future work.

## 2. Inverse Harmonic Oscillation

The forward problem of harmonic oscillation is well-studied. A simple example would be the forced harmonic oscillation of a spring-mass system. Given an external force  $f(t)$  as a function over time and other properties of the oscillating system, the forward problem seeks the response of the system, such as its displacement from the equilibrium position over time. Many natural motions can be seen as harmonic oscillation, such as the swaying of a plant in the wind or the movements of a boat due to regular water waves.

Our work requires solving the inverse problem to harmonic oscillation. As depicted in Figure 1, the only information we are given for the inverse problem is the observed displacement of the oscillator as well as some knowledge of the physical system being displaced. The goal is to recover the unknown parameters of the parameters of the external force and the natural phenomenon which caused the harmonic oscillation. There are three stages to our solution: scanning the displacement data, modelling the system, and synthesizing new animations. We now elaborate on the solution. The information flow is as shown in Figure 2 and the steps are as follows.

**Step 1:** Extract motion information from the video sequence.

**Step 2(a):** Use the oscillator’s displacement over time to estimate the natural frequency  $\omega_0$  and the damping coefficient  $\gamma$  of the oscillating system.

**Step 2(b):** Use the displacement data  $x(t)$  and the estimated information about the properties of the oscillating system to compute the external driving force  $f(t)$  which caused the motion.

**Step 2(c):** Use the estimated parameters of the driving

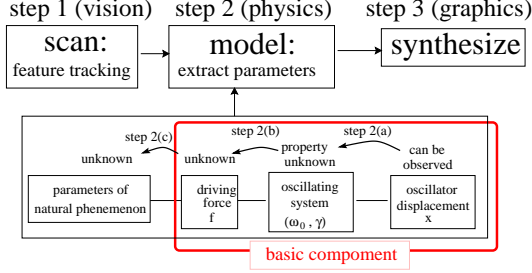


Figure 2: Solution schema for inverting harmonic oscillation.

force  $f(t)$  to infer the parameters of the natural phenomenon which produced the force.

**Step 3:** Once we have extracted the parameters of the natural phenomenon, we use these parameters to apply forces to computer generated objects.

The scanning and synthesis stages are dependent on the specific application. In Section 3, we describe the basic parameter extraction steps which are common to many applications of the inverse problem of harmonic oscillation, namely, steps 2(a) and 2(b) at the modelling stage in Figure 2. These two steps translate the observed displacement to the effective force parameters.

### 3. Oscillation Parameter Extraction

For an arbitrary function  $x(t)$  of time  $t$ , we will use the normal convention of denoting its first and second derivatives with to  $t$  as  $\dot{x}$  and  $\ddot{x}$ , respectively.

The relationship between the effective driving force  $f(t)$  over time  $t$  and the resulting displacement  $x(t)$  can be described using Newton's equation of motion:

$$\ddot{x} + \gamma\dot{x} + \omega_0^2 x = \frac{f}{m}, \quad (1)$$

where  $m\ddot{x}$  accounts for the inertial force,  $m\gamma\dot{x}$  is the frictional force,  $m\omega_0^2 x$  is the restoring force,  $\omega_0$  is the natural frequency and  $\gamma$  is the damping coefficient of the physical system's harmonic oscillation. Note that we divided Eq. 1 through by mass  $m \neq 0$  to isolate the unknown parameters.

In this section, we shall first briefly review of solutions to Newton's equation. For the inverse problem to harmonic oscillation, since we are not given the properties of the oscillating system, such as its natural frequency and damping coefficient, we will show in Section 3.2 how to estimate them. Using the estimated properties, we show in Section 3.3 how to recover the driving force parameters from observed displacement. We will demonstrate this process using an example.

### 3.1. Background: Solution to Newton's Equation

There are two classes of solutions to Eq. 1: one for forced oscillation when  $f \neq 0$ , and one for the oscillating system's transient behavior when  $f = 0$ . For the inverse problem to harmonic oscillation, the solution to the forced oscillation provides us with the necessary information about the properties of the oscillating system. So, here we will give a quick review of this type of solution. Interested readers are referred to [7, 8] for details.

A sinusoidal driving force at a particular frequency would cause a harmonic motion of the same frequency. In the case of forced harmonic oscillation, given a simple sinusoidal driving force

$$f(t) = f_i \sin(\omega_i t + \Delta_i), \quad (2)$$

the resulting displacement is a sine wave of the same frequency:

$$x(t) = \rho(\omega_i) f_i \sin(\omega_i t + \Delta_i + \theta_i), \quad (3)$$

where its amplitude is related to that of the driving force by a magnification factor  $\rho$ , and

$$\rho^2(\omega_i) = \frac{1}{m^2[(\omega_i^2 - \omega_0^2)^2 + \gamma^2\omega_i^2]}. \quad (4)$$

Its phase delay has an additional phase shift  $\theta$ , and  $\tan \theta = -\gamma\omega/\omega_0^2 - \omega^2$ .

In general, we can express the driving force in terms of its Fourier series expansion

$$f = f_0 + \sum_{i=0}^{\infty} f_i \cos(\omega_i t + \Delta_i),$$

where  $\omega_1$  is a fundamental frequency and  $\omega_i = i\omega_1$ , for  $i = 1, 2, \dots, \infty$ ,  $f_i$ 's are the Fourier series coefficients,  $\Delta_i$  are the phase delays. Since, for forced harmonic oscillation, the driving force at frequency  $\omega_i$  causes a displacement at the same frequency, one can calculate the displacement at each frequency using Eq. 3, then sum them up to obtain the resulting displacement

$$x = x_0 + \sum_{i=0}^{\infty} x_i \cos(\omega_i t + \Delta_i + \theta_i),$$

where

$$x_i = \rho(\omega_i) f_i. \quad (5)$$

### 3.2. Parameter Estimation

In order to extract the driving force parameters of a harmonic oscillator by only observing its displacement, we need to learn of its physical properties, including its natural frequency  $\omega_0$  and its damping coefficient  $\gamma$ . Harmonic

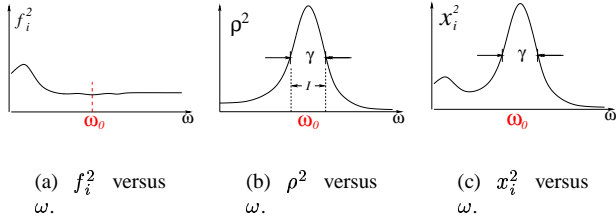


Figure 3: Relationship between the frequency domain response of the force and the resulting displacement.

oscillation is a resonance phenomenon, in which case information about  $\omega_0$  and  $\gamma$  are embedded in the displacement data. Intuitively, a harmonic oscillator tends to amplify forces near its natural frequency and annihilate forces at other frequencies. A closer inspection of Eqs. 3, 4 and 5 provides mathematical verification for our intuition.

Normally, if we look at the magnitude of an external force at different frequencies, we do not expect it to have special features near the natural frequency  $\omega_0$  of the oscillating system. In Fig. 3(a), we plot the square of the force amplitude  $f_i$  at frequency  $\omega = i\omega_1$ .

Equation 3 shows that the oscillating system acts as a filter where the magnitude of the force at frequency  $\omega_i = i\omega_1$  is amplified by a magnification factor  $\rho(\omega_i)$ . Figure 3(b) shows that  $\rho^2(\omega_i)$  peaks near the natural frequency  $\omega_0$  of the physical oscillating system, and it quickly decays away from  $\omega_0$ . Indeed, if we look at the length of the frequency interval  $\Delta\omega$  of the curve  $\rho^2(\omega_i)$  over which it achieves at least one-half of its maximum, it can be shown that for small  $\gamma$ ,  $\Delta\omega = \gamma$  [7]. This remarkable fact allows us to construct a good estimate for the damping coefficient  $\gamma$  from the observed displacement  $x_i^2$ . From Eq. 5, the magnitude of the displacement at frequency  $\omega_i$  is  $x_i = \rho(\omega_i)f_i$ . As shown in Fig. 3(c), if we plot  $x_i^2$  against  $\omega$  we would expect a significant peak near the natural frequency  $\omega_0$  of the physical oscillating system. Since by assumption there is no salient feature in an external force's frequency response near  $\omega_0$ , the width  $\Delta\omega$  over which the function  $x_i^2$  attains at least one-half its maximum height is a good estimate for  $\gamma$ .

In practice, we can sample the displacement function. Let  $\hat{x}(\tau)$  be the sampled displacement data at sampling point  $\tau$ , where  $\tau \in \{0, 1, \dots, N-1\}$ , total number of samples is  $N$ . the sampling period is  $T$ , and the sampling time is  $t = t_0 + \tau T$ . We examine  $\hat{x}$ 's behavior by transforming  $\hat{x}(\tau)$  to frequency domain using the discrete Fourier transform (DFT) [3]:

$$\hat{X}(\nu) = \frac{1}{N} \sum_{\tau=0}^{N-1} \hat{x}(\tau) e^{-j2\pi(\nu/N)\tau},$$

where the quantity  $\nu/N$  is analogous to frequency measured

in cycles per sampling interval, so the angular frequency is  $\omega = 2\pi\nu/(NT)$ . For  $i = 1, 2, \dots, \lfloor \frac{N}{2} \rfloor - 1$ ,  $x_i = 2|\hat{X}(\nu = i)|$ . So we can inspect  $|\hat{X}(\nu = i)|^2$  to estimate the peak at  $\omega_0$  and the damping coefficient  $\gamma$  in the same way as using  $x_i^2$ . An example estimation using real data (from Fig. 8a) is shown in Fig. 4.

### 3.3 From Displacement to Force

We shall use Eq. 5 to estimate the parameters of the driving force, Since the mass  $m$  of the observed object is unknown, Eq. 5 is rewritten as  $x_i = (m\rho(\omega_i))(f_i/m)$ , where  $m\rho(\omega_i)$  depends on only  $\omega_0$ ,  $\gamma$  and  $\omega$ . Hence we can estimate the parameter of the driving force  $f_i$  up to a scaling factor  $1/m$  at each frequency.

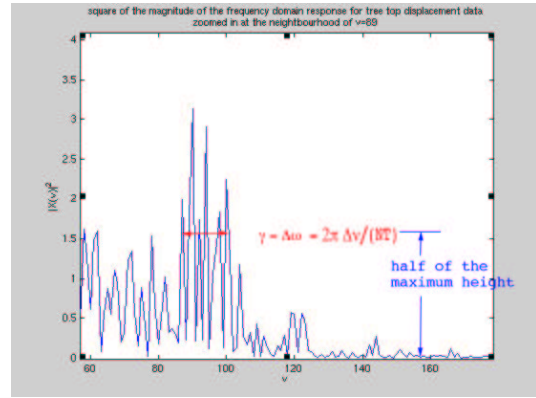


Figure 4: Estimate  $\gamma$  from  $|\hat{X}(\nu)|^2$ .

Since our tracking data is unlikely to be entirely accurate, we model the tracking error in  $\hat{x}(\tau)$  using Gaussian white noise [14, 17]. We performed white noise error analysis on our sample data. The power density spectra of the estimated parameters of the force and the expected noise in it are compared in Figure 5(Top). The ratio of the two power density spectra is plotted in Figure 5(Bottom). Because the signal to noise power density ratio is low at high frequencies, we cannot rely on estimated parameters in that band. The ratio is above 50 for much of the low frequency band in which  $\omega/\omega_0 \in [0, 1.2]$ , which gives us confidence that the force parameter estimation is satisfactory there. In this example, we choose to use only estimated parameters within this frequency range. Such frequency range varies for different example data.

Knowing the driving force parameter at each frequency, we can construct the driving force fluctuation  $f/m$  using a Fourier series. For our example data, in Figure 6, we compare the large scale fluctuation of the estimated driving force to a scaled version of the observed displacement. As we would expect, the large scale driving force fluctuation created the large scale movement of the observed object.

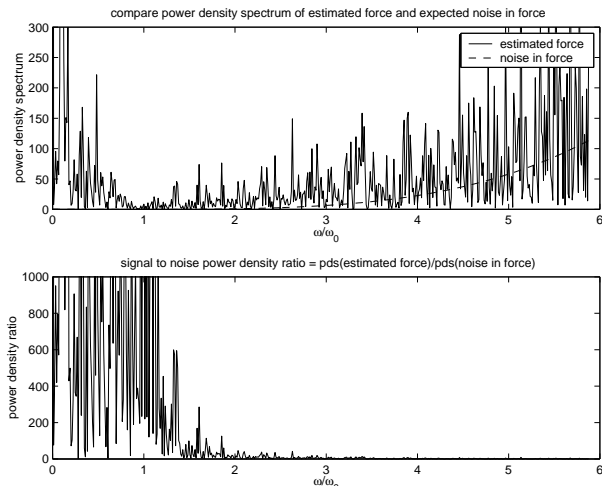


Figure 5: Noise in estimated force: (Top) power density spectrum of estimated force vs. expected PSD of noise in the force; (Bottom) power density ratio between estimated force and noise in the force.

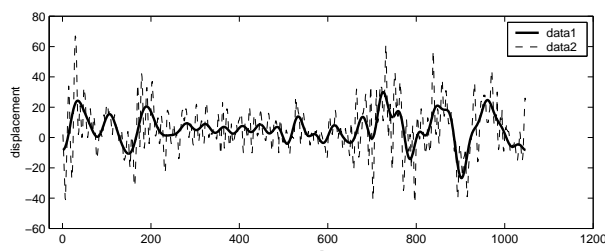


Figure 6: Compare the large scale driving force fluctuation (*data1*) against the observed oscillator displacement (*data2*).

## 4. Extracting Wind Speed

We wish to extract wind speed parameters by observing the motion of objects in the video such as plants and trees. To make the problem physically tractable, we will reduce the movement of the main branch of a tree to the movement of a simple cantilever beam. We first discuss the physical model. We define the cantilever beam model, followed by describing the relationship between an external force and the deflection of the beam. We discuss how the wind velocity produces a drag force. We then propose a solution to the scanning and extraction problem, and by way of example retarget extracted wind field parameters to the animation of synthetic snow and other plants within a real scene.

In Section 3, we explained the basic modelling component of the inverse problem to harmonic motion using a single degree of freedom (SDOF) system with a point mass to model an oscillator. In this section, we will use a cylindrical cantilever beam model to approximate flexible beams

like plants or a tree's main branch, as shown in Figure 7(b). We will show the close relationship between the cantilever beam model and the point mass model. We will use our knowledge of the point mass model to solve for the oscillation parameters of the cantilever beam model.

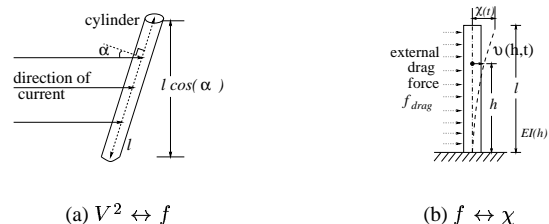


Figure 7: (a) Project the length of the cylinder to a plane perpendicular to the direction of the current. (b) Flexure beam treated as a SDOF system.

### 4.1. Beam Bending: Force and Deflection

When wind flows past a beam, it is subjected to an external drag force  $f_{drag}$  perpendicular to it. We define a relationship between the displacement at the top of the beam and the drag force. Consider the formulation of an equation of motion for the cantilever beam in Figure 7(b) in two dimensions (2D), the essential properties of the beam (excluding damping) are listed below.

$l$	the length of the beam
$h = 0 \dots l$	the height of a point on the beam
$m_{total}$	the mass of the beam

To approximate the motion of this system with a single degree of freedom, it is necessary to assume that the beam will deform only as a single shape. The shape function will be designated  $\varphi(h)$ , and the amplitude of the motion relative to the moving base will be represented by the generalized coordinate  $\chi(t)$ . Thus, the shape of the beam at a particular time  $t$  is  $v(h, t) = \varphi(h)\chi(t)$ . For the shape of the bending beam, there are a number of basic shapes often used in engineering texts. For our example, our choice is based on a beam shape equation in [26]:  $\varphi(h) = \frac{h^2}{3l^2}(6l^2 - 4hl + h^2)$ . The maximal deflection at the top of the beam is  $\varphi(l) = 1$ , and hence  $\chi(t) = x(t)$ . The equation of motion of this generalized SDOF system can be formulated using the principle of virtual work. Interested readers are referred to [6, 25, 26] for details.

$$\ddot{x}(t) + \gamma^* \dot{x}(t) + \omega_0^{*2} x(t) = c_{shape} f_{drag} / m_{total}, \quad (6)$$

where  $c_{shape} = \int_0^l \varphi(h) dh / \int_0^l \varphi^2(h) dh = 81/52$  is a constant. Applying the technique we discussed in section 3 to the beam top deflection  $x(t)$ , we can estimate the drag force within to a scaling factor of  $c_{shape} / m_{total}$ .

## 4.2. Wind Velocity and Drag Force

When wind flows around a cylinder, it induces a drag force  $f_{drag}$  that is proportional to the square of the wind velocity  $V^2$ :

$$f_{drag} = \frac{V^2}{C_D \frac{1}{2} \sigma D l \cos \alpha}, \quad (7)$$

where  $C_D$  is the drag coefficient,  $D$  is the diameter of the cylinder,  $l$  is the length of the cylinder,  $\sigma$  is the density of the fluid, and  $l \cos \alpha$  is the projection of the cylinder length to the plane perpendicular to the direction of the liquid (see Figure 7(a)). Engineers typically treat the drag coefficient as a constant, such as  $C_D = 1.2$  [19, 7]. Since we can estimate the drag force within a scaling factor, we can estimate the square of the wind velocity within a scaling factor.

## 4.3. Tracking Branches

Our implementation of the automatic tracking of features in video streams is based on work by Jepson, Fleet and El-Maraghi [10]. For example, to estimate the wind speed we track the movement of the top of the tree in Figure 8(a). The regions in the ellipses are tracked from frame to frame. To synthesize computer generated objects, we normally need only the tracking result from one of the ellipses. For instance, in this example, we pick the rightmost ellipse. The center of the ellipse is used as an approximation for the top of the tree branch. We record the dis-

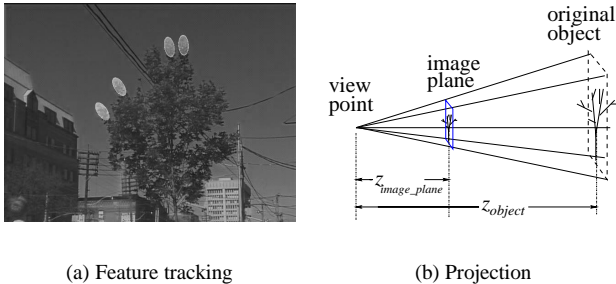


Figure 8: (a) Tracing the movement of the top of tree branches. (b) The relationship between  $\hat{x}_{image\_plane}$  in the image coordinate system and the corresponding  $x$  in 3D space.

placement ( $\hat{x}_{image\_plane}, \hat{y}_{image\_plane}$ ) of the branch top from its resting position in the image coordinate system. The resting position of the top of the tree branch is estimated by the user. During the video sequence, there are periods of time when the wind dies down, and the tree branch comes to rest briefly. Since, most of the time, the change in the  $y$  component is negligible, we only need to consider the displacement in the  $x$  component. Since the displacement data is sampled at time interval  $T$ , we will use the notation  $\hat{x}_{image\_plane}(\tau)$ , where sample index

$\tau \in \{0, 1, \dots, N - 1\}$ , and  $N$  is the total number of samples. Assuming that the motion is roughly planar with a small relative depth variation, the measured displacement value  $\hat{x}_{image\_plane}$  in the image coordinate system is proportional to the corresponding displacement  $x$  of the real object (see Figure 8(b)). This factor can be absorbed in the unknown proportionality constant relating the observed displacements and the estimated wind speed.



Figure 9: (a) Adding a computer synthesized shrub in the scene and let it sway along side of the trees in the video sequence. (b) Using the estimated wind velocity parameters to drive the falling snow animation.

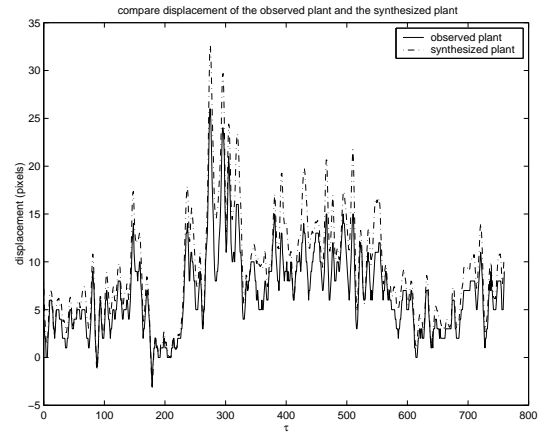


Figure 10: Comparing the displacement of the synthesized plant to the displacement of an observed plant in Fig.9a.

## 4.4. Synthesis Examples

The analysis above was applied to tracking data from the two sequences shown in Fig. 9. The wind speed was estimated up to a constant multiplicative factor, and then reapplied to drive synthetic objects. In Fig. 9a the estimated wind was applied to a synthetic shrub. The synthesized displacement of one branch of the shrub is shown in Fig. 10, and the similarity in the overall structure of the synthesized displacement with the observed displacement indicates that

our procedure has successfully captured the dominant effects of the wind forces. In Fig. 9b we use the estimated wind to drive synthetic snow.

## 5. Bobbing Boats

A ship moving on the surface of open water is almost always in oscillatory motion [2]. There are six different kinds of motions that a ship experiences: three linear and three rotational about its three principal axes. Only three kinds of motion, namely, heaving, rolling, and pitching, are purely oscillatory motions, since these motions act under a restoring force or moment when the ship is disturbed from its equilibrium position. Currently, we are restricting our parameter extraction experiments to scenarios where the forcing function is due to the water waves interacting with the hull of the ship and, moreover, we assume the direction of the water wave is parallel to either the longitudinal axis or the transverse axis of the boat.

In [24] it is shown that the equations of motion for the pitching angle  $\alpha(t)$  of a boat can be approximated by the harmonic oscillator

$$\ddot{\alpha} + \gamma_p \dot{\alpha} + \omega_{0p}^2 \alpha = M_p / I_p, \quad (8)$$

where  $I_p$  is the mass moment of inertia of the boat about its transverse rotational axis,  $\gamma_p$  is the damping coefficient and  $\omega_{0p}$  is the natural frequency for pitching. The forcing term  $M_p(t)$  is the exciting moment which is a function of the wave height in the neighbourhood of the boat's hull (see [24] for the details). Similar equations can be derived for rolling and heaving motions.

### 5.1. Tracking Rolling and Pitching

We observed sailboats anchored by the shore. We video taped a sailboat with dominating rolling motion with the forward axis of the camera perpendicular to the rolling motion. Since the harbor is a relatively sheltered environment, heaving motion is not prominent enough to be analyzed.

For a rolling or pitching video sequence, we decide on a sailboat to be tracked. We wish to monitor the inclination of a sailboat by tracking two points of the its mast – the top of the mast  $P_1$  and the base of the mast  $P_2$ , as illustrated in Figure 11(b). Figure 11(a) shows one frame of the tracking result. The centers of the ellipses are the points tracked by the tracking software.

We assume the image's vertical axis is parallel to the upright position of the mast of the sailboat. So the line  $\overline{P_1 P_2}$  and the image's vertical axis form an angle  $\alpha$ , which is the inclination of the sailboat in 2D. When the boat is experiencing a pitching dominated motion, we video taped it with the camera's forward axis perpendicular to the pitching motion, in which case  $\alpha$  would be the pitching inclination. Similarly, for rolling, the camera setup would ensure that  $\alpha$  is the rolling inclination.

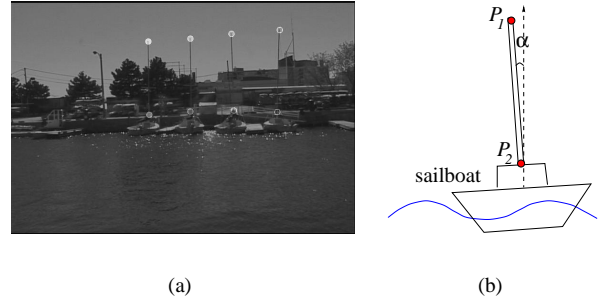


Figure 11: (a) One frame of the tracking result. (b) Sailboat rolling or pitching angle.

For our purposes, we do not need to determine the equilibrium position for rolling or pitching, because it is merely a constant which contributes to the parameter at angular frequency  $\omega = 0$ . Intuitively, the water wave corresponding to  $\omega = 0$  is flat, and it does not produce turning moment for the boat.

### 5.2. Rolling and Pitching Synthesis

Given simple assumptions about the shape of the hull, the estimated pitching moment  $M_p(t)$  can be related to the variation of wave height over time, modulo an unknown multiplicative constant. In fact, in naval architecture, engineers often work with very simple geometric models during the design stage, in order to have an intuitive rough estimate of the dynamic properties of the marine vehicle. For our purpose, users may use their intuition to give a rough estimate for the length, width and height of the rectangular block model of the observed boat. We can use this rough model to infer water wave elevation parameters. These parameters can then be used to produce pitching and heaving motion for synthetic boats to give the visual effect of having the observed boat and the virtual boats moved by the same water waves.

This analysis-synthesis process has been applied to the generation of synthetic pitching and rolling motions (see Fig. 12). Using our method, we can easily introduce many computer synthesized boats in the scene. The motions of boats are of course not completely unrelated, because they are affected by the same water wave.

## 6. Conclusion

We have proposed a novel way to indirectly drive computer animations using real video input, calling it Video Input Driven Animation (VIDA). We observe the motion of objects in the video captured scene, extract and estimate useful parameters, and re-apply these parameters to drive the animation of computer generated objects. An example ap-

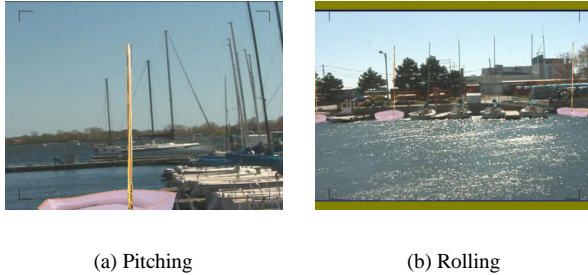


Figure 12: (a) Sample animation frame from adding the computer synthesized boat to a scene with pitching boats and animating the new boat's pitching motion. (b) Putting computer synthesized boats in the scene and letting them move alongside of the video captured sailboats.

plication of this idea would be to use real life video as a background, and make computer generated objects appear as natural participants in the video. We demonstrated the concept by developing a model for inverse harmonic oscillation and using it to extract parameters for two different phenomena: wind and regular water waves.

One limitation of VIDA is due to the limited resolution of our video camera. When an object of interest is far away, its movement may span merely a few pixels. In this case, measuring accuracy becomes very critical, and we might not be able to capture enough details of the objects motion to make the analysis meaningful. Another limitation is that currently we are not considering the situation in which the foreground computer synthesized object might create large alterations of the background environment. This situation requires further study.

For computer graphics, VIDA introduces a rich source for animation creation and animation control. It allows us to use video as a way for a user to specify indirect inputs, such as forces and constraints, to a computer animation. VIDA poses many new research challenges for both computer graphics and computer vision.

## References

- [1] K. S. Bhat, S. M. Seitz, J. Popović, and P. Khosla. Computing the physical parameters of rigid-body motion from video. *Proc. ECCV*, pages 551–566, 2002.
- [2] Bameswar Bhattacharyya. *Dynamics of Marine Vehicles*. John Wiley & Sons, 1978.
- [3] Ronald N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill Book Company, second edition, 1983.
- [4] Matthew Brand. Voice puppetry. *Computer Graphics*, pages 21–28, August 1999.
- [5] B. R. Clayton and R. E. D. Bishop. *Mechanics of Marine Vehicles*. E. & F. N. Spon Ltd, 1982.
- [6] Ray W. Clough and Joseph Penzien. *Dynamics of Structures*. McGraw-Hill, Inc., second edition, 1993.
- [7] Richard Phillips Feynman. *The Feynman lectures on physics*, volume 1–3. Addison-Wesley Pub. Co., Reading, Massachusetts, 1963.
- [8] A. P. French. *Vibrations and Waves*. W. W. Norton & Company, New York, 1971.
- [9] Thomas C. Gillmer and Bruce Johnson. *Introduction to Naval Architecture*. Naval Institute Press, 1982.
- [10] Allan D. Jepson, D. J. Fleet, and T. El-Maraghi. Wsl: Robust online appearance models for visual tracking. *Proc. IEEE CVPR*, pages 415–422, 2001.
- [11] Nina Morgan. *Marine Technology Reference Book*. Butterworth & Co. (Publisher) Ltd, 1990.
- [12] W. Muckle and D. A. Taylor. *Muckle's Naval Architecture*. Butterworth & Co. (Publisher) Ltd, 1987.
- [13] Minoo H. Patel. *Dynamics of Offshore Structures*. Butterworth & Co. (Publisher) Ltd, 1989.
- [14] Peyton Z. Peebles, Jr. *Probability, Random Variables, and Random Signal Principles*. McGraw-Hill Book Company, second edition, 1987.
- [15] Zoran Popović and Andrew Witkin. Physically based motion transformation. *Computer Graphics*, pages 11–20, August 1999.
- [16] Tim Poston and Ian Steward. *Catastrophe Theory and its Applications*. Pitman Publishing Limited, 1978.
- [17] John G. Proakis and Dimitris G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Macmillan Publishing Company, second edition, 1992.
- [18] Henry E. Rossell and Lawrence B. Chapman. *Principles of Naval Architecture*. The Society of Naval Architects and Marine Engineers, 1942.
- [19] Hermann Schlichting. *Boundary Layer Theory*. McGraw-Hill Book Company, Inc., fourth edition, 1960.
- [20] Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. In *Proceedings of ACM SIGGRAPH 2000*. Computer Graphics Proceedings, Annual Conference Series, pages 489–498. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, July 2000. ISBN 1-58113-208-5.
- [21] Mikio Shinya and Alain Fournier. Stochastic motion – motion under the influence of wind. *Eurographics '92*, 11(3):119–128, 1992.
- [22] Mikio Shinya, Takeaki Mori, and Noriyoshi Osumi. Periodic motion synthesis and fourier compression. *The Journal of Visualization and Computer Animation*, 9:95–107, 1998.
- [23] Jos Stam. Stochastic dynamics: Simulating the effects of turbulence on flexible structures. *EUROGRAPHICS'97 Proceedings*, 16(3):159–164, 1997.
- [24] Meng Sun. *Video Input Driven Animation (VIDA)*. PhD thesis, University of Toronto, 2003.
- [25] S. Timoshenko and J. M. Lessells. *Applied Elasticity*. Westinghouse Technical Night School Press, East Pittsburgh, PA, first edition, 1925.
- [26] S. Timoshenko and Gleason H. MacCullough. *Elements of Strength of Materials*. D. Van Nostrand Company, Inc., Princeton, New Jersey, third edition, 1949.