Motion from Transient Oscillations

Jeffrey E. Boyd
Department of Computer Science
University of Calgary
Calgary AB T2N 1N4
boyd@cpsc.ucalgary.ca

James J. Little
Department of Computer Science
University of British Columbia
Vancouver BC V6T 1Z4
little@cs.ubc.ca

Abstract

Evidence exists to suggest that mechanisms humans use to perceive motion are related to the mechanisms humans use to produce motion. This requires that humans perceive and produce complex sets of oscillations. These observations provide the motivation for video phase-locked loops (vPLLs) which synchronize internal oscillators to oscillations in an image. Observations of vPLLs in a real-time system reveal they are also sensitive to translational motions because it synchronizes to the transient response of input filters. This paper compares optical flow computed as a byproduct of vPLL synchronization with conventional optical flow algorithms. While vPLLs are not likely to replace conventional optical flow algorithms, their performance demonstrates that a single system can be sensitive to both oscillations and translations, which could be useful in motion recognition systems.

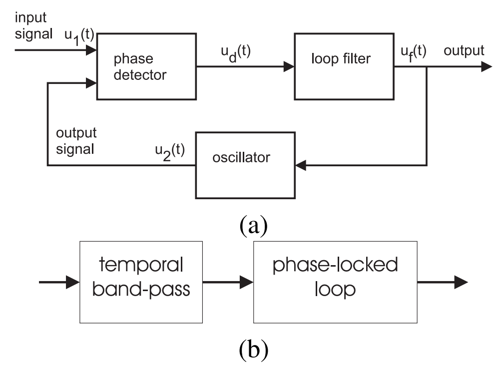

Figure 1. The vPLL: (a) a basic PLL and (b) a PLL with a temporal band-pass filter at the input. In a vPLL, there is one PLL/band-pass filter combination at each pixel site.

1. Introduction

Fazio et al. [5] suggest, based on PET data, that the parts of the human brain used to perceive certain human motion are the same parts used when a human produces that motion. Bertenthal and Pinto [2] present psychophysical evidence that suggests a connection between synthesis of motion and its perception by humans. It is evident from efforts to synthesize human gaits [10, 1, 7] that synthesis requires production of a set of complex, well-coordinated oscillations.

These observations provided the motivation for developing video phase-locked loops (vPLL) [3]. The vPLL uses an array of phase-locked loops (PLL) shown schematically in Figure 1(a). A band-pass filter tuned to the locking range of the PLL at each pixel restricts the PLL inputs to oscillations in the frequency range that can be locked, Figure 1(b). A PLL uses a control loop to synchronize the oscillations of an internal oscillator to an external oscillation applied at its input. External oscillations are primarily due to observation of a gait. Thus the vPLL is an example of a system that perceives an external oscillating stimulus by matching it with internal oscillations. Liu and Picard [8] demonstrate a system that uses Fourier transforms to extract similar information from gait sequences.

Although the vPLL is well-suited to analyzing oscillating motions, its use in a real-time system reveals that vPLLs exhibit a response to translational motions too. Initially this appears to be a curious side effect, but reveals an interesting property of the system. Translational motion causes pixels to see transient signals at a pixel. This causes the PLL in the pixel to synchronize to the transient response of the band-pass filter. The effect gives the vPLL the ability to compute something similar to optical flow. Therefore, vPLLs show that it is possible to perceive both translation and oscillation with a single system.

A familiar formula relates velocity to distance and time: $v = \Delta x / \Delta t$, where v is velocity, x is distance, and t is time. There are, therefore two requirements to compute a velocity: a measure of *distance*, and a measure of *time*. The following sections look at conventional optical flow algorithms and vPLLs highlighting how the algorithms meet these two

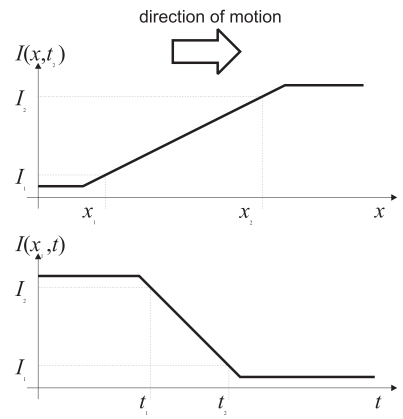

Figure 2. Signals extracted from a hypothetical temporal image sequence to illustrate gradient flow methods.

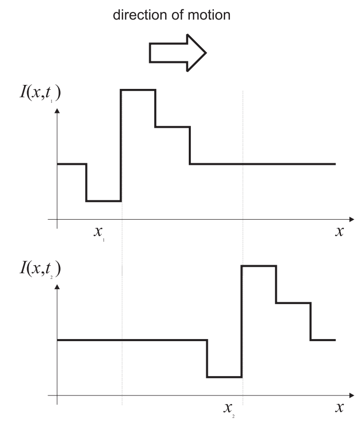

Figure 3. Signals extracted from a hypothetical temporal image sequence to illustrate region-matching flow methods.

requirements.

2. Conventional Optical Flow

Gradient Methods: Figure 2 shows a set of signals extracted from a hypothetical temporal image sequence, reduced to one spatial dimension. The following equations summarize the computation of optical flow from the signals:

$$\begin{array}{cc} \Delta x = x_2 - x_1 & \Delta I = I_2 - I_1 \\ I_x \approx \frac{\Delta I}{\Delta x} & I_t \approx -\frac{\Delta I}{\Delta t} \\ \multicolumn{2}{c}{v = \frac{\Delta x}{\Delta t} \approx -\frac{I_t}{I_x}} \end{array}$$

where I_x and I_t are partial derivatives of I, the image intensity, with respect to x and t respectively, and v is the flow. This reduces to $I_x v + I_t = 0$, which is a one-dimensional variation of the well-known optical flow constraint equation [6] used in gradient methods.

Gradient algorithms meet the distance- and time-measurement requirements by measuring the partial derivatives I_x and I_t. This assumes that the derivatives are constant where the flow is computed. While this is not generally true, in local image regions and consecutive images in a sequence, it is a reasonable assumption.

Region-Matching Methods: Figure 3 shows a set of signals extracted from a hypothetical temporal image sequence, reduced to one spatial dimension. To compute optical flow, a region-matching flow algorithm matches the signal $I(x, t_1)$ to $I(x, t_2)$. t_1 and t_2 may be selected arbitrarily but are typically associated with consecutive images [4]. The matching algorithm determines the spatial locations x_1 and x_2 that match. Thus optical flow may be computed using

$$v = \frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{t_2 - t_1}.$$

Region-matching flow algorithms meet the distance-measurement requirement by locating matching regions in images. The time-measurement requirement is met by the selection of images in a temporal sequence that are matched.

3. Video Phase-Locked Loops

Figure 4 shows a set of signals extracted from a hypothetical temporal image sequence, reduced to one spatial dimension. The figure shows a step profile that passes over an image region. Each pixel sees a temporal step function, indicated with a dashed line on the temporal axes, $I(x, t)$. The band-pass filter at each pixel responds to the step with the filter's step response, $G(x, t)$. This is the damped oscillations shown as a solid line in the figure. The PLL responds to the oscillations by synchronizing its internal oscillator to $G(x, t)$.

The frequency and the phase of the oscillations in $G(x, t)$ give a measurement of time. In effect, the step response and PLL combination act as a timer. The timer begins timing when the step arrives at the pixel and continues until the step response oscillations are damped.

To compute flow, select two points in the image at positions x_1 and x_2, and compare the measured phases of the band-pass outputs, ϕ_1 and ϕ_2. The flow is given by

$$\begin{aligned} \Delta x &= x_2 - x_1 \\ \Delta \phi &= \phi_1 - \phi_2 = \omega \Delta t \\ v &= \frac{\Delta x}{\Delta t} = \frac{\omega \Delta x}{\Delta \phi} \end{aligned}$$

where ω is the frequency of $G(x, t)$. The above equation is valid only when there is motion, i.e., when the magnitude

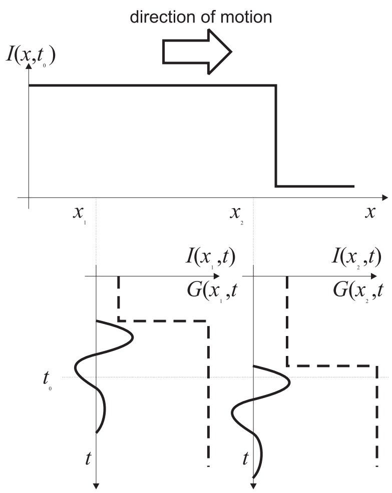

Figure 4. Signals extracted from a hypothetical temporal image sequence to illustrate vPLL motion computation.

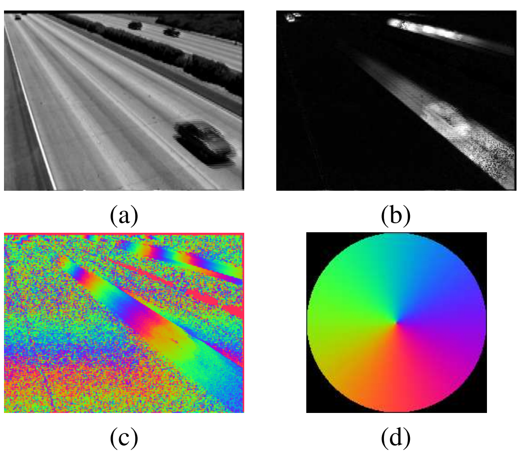

Figure 5. Example of vPLL applied to a translation: (a) original sequence, (b) the magnitude output, (c) the phase output, and (d) color coding scheme for the phases. MPEG coded sequences can be found at /www.cpsc.ucalgary.ca/˜boyd/techsketch

of the oscillations, $|G|$ exceeds a threshold, and when $\Delta\phi$ is non-zero. Thus the validity of a flow measurement is given by

$$\text{valid} = \begin{cases} \text{true} & |G| > g_{\min} \text{ and } \Delta\phi > \Delta\phi_{\min} \\ \text{false} & \text{otherwise} \end{cases} .$$

Figure 5 shows an example of a vPLL detecting the translational motion of the vehicles on a freeway. The magnitude response is shown in Figure 5(b). The bright regions in the image highlight the fact that there is motion present at those points in the image. Phase information, Figure 5(c) encodes the direction and speed of the motion. Along the path of a car one can see the cycling of phases produced by the vPLL locking to the transient responses of the band-pass filters.

The locking time of the PLLs affects the output. Depending on the state of the PLL when the step occurs at the pixel, the PLL can lock in anywhere from zero to 40 or 50 frames. This means there is a delay between when the step occurs and when the motion may be reliably detected at the pixel. This delay is variable, and almost random. Once the PLL has locked on the oscillations, the motion estimate becomes reliable. This can be seen in Figure 5(c) where the phase data is only stable at the trailing edge of the vehicle.

Example calculation: In frames 40 and 50 the front bumper of the car is at positions $x_1 = (85, 41)$ and $x_2 = (102, 54)$ respectively. Thus $v = (1.7, 1.3)$pix/frame, determined manually. If we look at the phase in these two positions after the vPLL has locked, at frame 60, we find $\phi_1 = 1.2$rad, $\phi_2 = 4.9$rad, and $\Delta\phi = 2.6$rad. By design, $\omega = 2\pi(0.05625)$. Therefore, using vPLL data we get the estimate $v = (2.3, 1.8)$. For comparison, the Lucas and Kanade [9] algorithm (LK) gives $v = (1.75, 1.25)$, while the Bulthoff Little and Poggio [4] algorithm (BLP) gives $v = (2, 1)$. Similarly, in frames 80 and 85 the front bumper of the car is at positions $x_1 = (237, 151)$ and $x_2 = (291, 192)$ respectively, giving $v = (10.8, 8.2)$. The vPLL gives $\phi_1 = 3.2$rad, $\phi_2 = 1.8$rad, $\Delta\phi = 1.4$rad, and $v = (13.8, 10.5)$. For comparison, LK gives $v = (0, 0)$, while BLP gives $v = (18, 15)$. The conventional algorithms have difficulty due to the large flow and foreshortening distortions.

Figure 6 shows a vPLL detecting a repeating translation. In this case the repetition means the image is actually oscillating and the vPLL locks to these oscillations. The figure shows phasors (vectors showing the magnitude and phase of sinusoids at a common frequency) representing the oscillations extracted by the vPLL. Like the example in Figure 5, the phases cycle through $-180°$ to $180°$ along the path of the apparent motion, indicating a translation. The repetitions force the vPLL to lock to the frequency of repetition rather than the frequency of the band-pass filter step response.

Example calculation: By design, the flashing lights *move* 280 pixels over 21 frames giving $v = 13.3$pix/frame. The frequency of repetition is $2\pi/24$ which is locked by the vPLL. The phasor plot indicates that $\Delta\phi \approx \pi/4$ for any pair of adjacent lights, and adjacent lights are 40 pixels apart. Thus the vPLL flow estimate is also $v = 13.3$pix/frame. It worth noting that some conventional flow algorithms have trouble with this sequence. LK fails because the assumption of constant gradient is violated by the large spacing of the lights. BLP computes flow but compares only consecutive

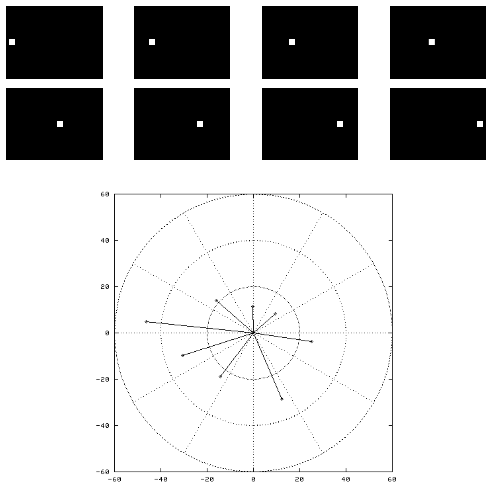

Figure 6. Image sequence showing illusion of motion created by sequentially flashing lights (top), and phasors extracted by a vPLL at the light positions (bottom). MPEG coded sequences can be found at /www.cpsc.ucalgary.ca/˜boyd/techsketch

frames. Since the light is stationary for 2 frames and then jumps 40 pixels it estimates zero flow for two frames and 40 pix/frame for the third. The mean flow is correct, but BLP does not see this.

4. Discussion

The selection of pixels to compute vPLL flow is arbitrary, although the pixels must be sufficiently close to avoid phase wrapping. Adjacent pixels in either the x or y direction allow computation of flow field as is done with conventional algorithms. Alternatively, some prior knowledge of the scene can be used. For example, in the freeway scene of Figure 5, pixels chosen along the lanes can be used to measure the speed of cars.

While it is fair to conclude that vPLLs can compute flow, it is not likely to compete with known flow algorithms to produce a motion field. The variability in the magnitude response and the delay in locking are major difficulties. This may change with further development of vPLLs.

vPLL-derived flow relies on the step-response properties of a band-pass filter to measure time between step events. In this sense vPLL flow is an artifact of the band-pass filter. However, all flow algorithms use some artifact of data or algorithm to produce a measure of time. The band-pass filter and PLL are merely less obvious and rely on oscillations.

If vPLL-derived flow is not likely to replace known optical flow algorithms then why is it interesting? We give the following answers to this question:

1. If we accept the PET and psychophysical evidence then human perception of motion requires a mechanism to deal with oscillations. It would be convenient (although not necessary) to use the same mechanism to perceive both translations and oscillations. vPLLs show that such a mechanism can exist, and that two distinct mechanisms may not be necessary.

2. When a vPLL is observed operating in real-time, the response to translations is obvious. It is satisfying to find an explanation of this surprising ability of vPLLs.

References

[1] K. Amaya, A. Bruderlin, and T. Calvert. Emotion from motion. In *Graphics Interface 96*, 1996.

[2] B. I. Bertenthal and J. Pinto. Complementary processes in the perception and production of human movements. In L. B. Smith and E. Thelen, editors, *A Dynamic Systems Approach to Development: Applications*, pages 209–239. MIT Press, Cambridge, MA, 1993.

[3] J. E. Boyd. Video phase-locked loops in gait recognition. In *International Conference on Computer Vision*, 2001.

[4] H. Bulthoff, J. J. Little, and T. Poggio. A parallel algorithm for real-time computation of optical flow. *Nature*, 337:549–553, February 1989.

[5] F. Fazio, D. Perani, N. A. Borghese, S. Ferrari, F. L. Monaco, V. Blasi, and M. C. Gilardi. How can virtual reality be? a pet study of motor actions observation. In *Human Brain Mapping Conference*, 1999.

[6] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, Massachusetts, 1986.

[7] J. Laszlo, M. van de Panne, and E. Fiume. Limit cycle control and its application to the animation of balancing and walking. In *SIGGRAPH 96*, pages 155–162, 1996.

[8] F. Liu and R. W. Picard. Finding periodicity in space and time. In *International Conference on Computer Vision*, 1998.

[9] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[10] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *SIGGRAPH 95*, pages 91–96, 1995.