Detecting hand-ball events in video sequences

Nicholas Miller
nmiller@uwaterloo.ca

Richard Mann
mannr@uwaterloo.ca

David R. Cheriton School or Computer Science
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Abstract

We analyse video sequences of a hand interacting with a ball. The hand motion is segmented using a piecewise polynomial motion model inspired by research in motor control. Next, an adaptive gravitational model of the ball is used to locate hand-ball events. We show that hand-ball events can be automatically classified from velocity and acceleration profiles.

1. Introduction

Given the trajectories of moving objects, what physically meaningful aspects of the motion can be recovered?

We consider the problem of interpreting ballistic and non-ballistic motion in real video sequences, such as shown in Fig. 1. In this video the hand manipulates the ball – first, by carrying it, then throwing, and finally catching it after the ball bounces on the wall and floor.

Our eventual goal is to characterize events based on *qualitative scene dynamics*. For example, given the above video we should infer that an "active" hand is moving a "passive" ball by applying a force. Once released, the ball is undergoing (passive) gravitational motion as it moves through the air and bounces off the wall. In [6] a system was presented that infers scene dynamics based on the Newtonian mechanics of a simplified scene model. However, that system was limited to the instantaneous analysis of continuous motion. Sequences were processed on a frame by frame basis, and discontinuous motions (due to contact changes, collisions, or starts and stops of motion) were explicitly removed. To apply dynamics analysis to extended sequences, we require a way to identify the motion boundaries, and to determine the precise velocity and accelerations at such boundaries.

In previous work [5, 4], we showed that ballistic motion can be accurately described using a piecewise quadratic motion model. The sequences were segmented using dynamic programming, and motion boundaries were then classified based on their velocity and/or acceleration discontinuities. In that work, neither hand motion nor non-ballistic ball motion were considered.

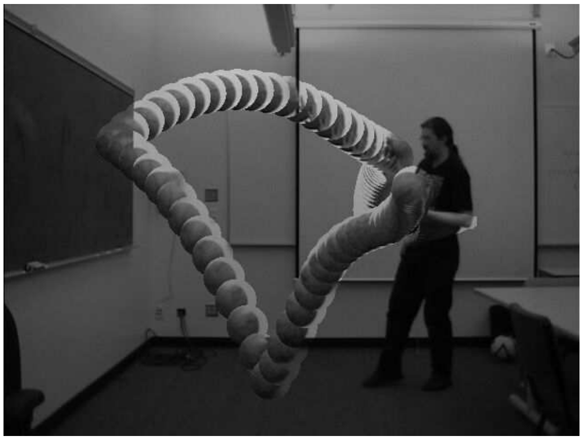

Figure 1. A composite of the tracking results for a sequence where a subject throws a basketball.

The goal of this work is to analyze the motion of the ball under non-ballistic motion, and in particular, motion caused by the hand. Fig. 2 shows the trajectories of the hand and ball overlaid on the image frame. The lines go from oldest (grey) to most recent (black). The circles show the segmentation of the hand trajectory based on a piecewise polynomial model. Our analysis of ball motion is based on three observations. First, we can determine the event type (catching, releasing, or hitting) based on the hand proximity. Second, given the event type, we can fit an adaptive gravitational model to the ball to determine the precise extent of gravitational and non-gravitational motion. Finally, to classify the non-ballistic motion, we compare the velocity of the ball and the velocity of the hand *at the motion boundary*. Fig. 3 shows the classification of the resulting motion boundaries and their corresponding frames in the videos. The arrows in the lower left corner of each panel show the

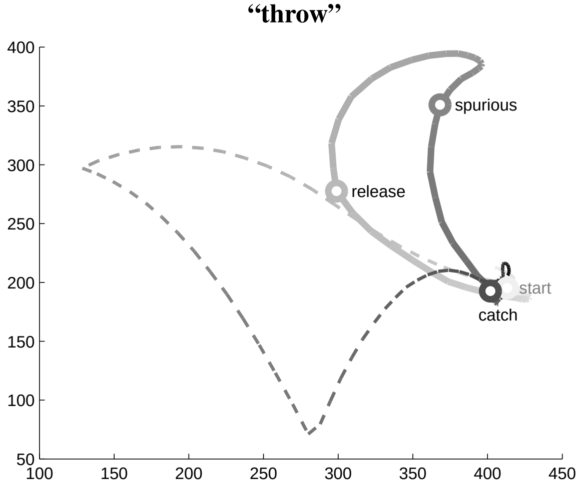

Figure 2. Hand (thick lines) and ball (thin lines) trajectories for the "throw" motion sequence. Time goes from oldest (grey) to newest (black). Circles show hand segmentation found in Sec. 2.2.

steps in velocity and acceleration that may be used to classify the events.

This work makes three contributions. First, we show that hand motion can be segmented using piecewise fifth order polynomials inspired by work in motor control [1]. Second, by fitting a context-dependent gravitational model to the ball over an adaptive window, we can isolate places where the hand is causing non-gravitational motion of the ball. Finally, given precise segmentation, we use the measured velocity steps (force impulses) on the ball to classify various event types.

2 Hand segmentation

The video sequences in Fig. 3 were captured with a consumer camcorder (Canon Optura Pi) at 640x480, 30fps, non-interlaced. An adaptive view-based tracker [3] was used to track the ball (circle) and forearm (elongated octagon), shown as highlighted parts in the frames. The motion was roughly parallel to the image, so the scene depth was relatively constant. Hence we can safely treat a two dimensional model for the projection of the hand and ball as a true model for the hand and ball (a weak perspective model). For subsequent trajectory processing, we use the center of the circle for the ball and the endpoint of the octagon for the hand.

2.1 The Minimum Jerk Principle

An explicit piecewise smooth model of the hand trajectory in a fixed canonical co-ordinate frame was necessary. The natural approach to devising an appropriate model is to start with a simple universal principle which could give rise to the varieties of intentional hand motions we regularly encounter. Such a model was previously proposed in the Psychological literature concerning motor control. Flash and Hogan [1] devised and verified a model based on the principal of minimizing the integral of the square of the magnitude of the jerk (which is defined as the rate of change of the acceleration of the hand). Using the calculus of variations they determined that unconstrained hand motion can be described by fifth order polynomials in time. This principal is known as the minimum jerk principle [1].

Fifth order polynomials also provide some desirable properties for our purposes. They are continuous and smooth and it is easy to calculate their derivatives which we can use to estimate velocities and accelerations. They are the minimum degree polynomial that can describe one smooth continuous motion which is constrained to have zero acceleration and velocity at the endpoints like a typical hand movement. Also we are able to use the least squares technique to fit a polynomial to the hand position data and perform the segmentation algorithm described below.

It was necessary to determine the coefficients of the fifth order polynomials in a slightly different way than did Flash and Hogan [1] when they were verifying their model. They would impose given position, velocity, and accelerations end points (and in some cases, an interior position point) of one single hand motion. This uniquely specifies the fifth order polynomial describing the hand trajectory between the end points. The video sequences do not, however, contain pre-specified motion endpoints. In fact we found that imposing arbitrary endpoints can give rise to some unnatural trajectory predictions (including cusps and loops) which do not correspond to the hand data. We use the following approach.

2.2 Dynamic Programming

We consider the segmentation of the hand motion into piecewise fifth order polynomial segments. The total cost of a segmentation is given by

$$\text{Cost} = \sum_{n=1}^{N} \left[\sum_{t=t_{n-1}}^{t_n} \left\| \mathbf{X}(t) - \hat{\mathbf{X}}_n(t;\theta_n) \right\|^2 + \lambda \right] \qquad (1)$$

where $\mathbf{X}(t)$ is the observed hand motion, $\hat{\mathbf{X}}_n(t;\theta_n)$ is the nth polynomial segment with polynomial coefficients θ_n which we use to estimate the hand's velocity and acceler-

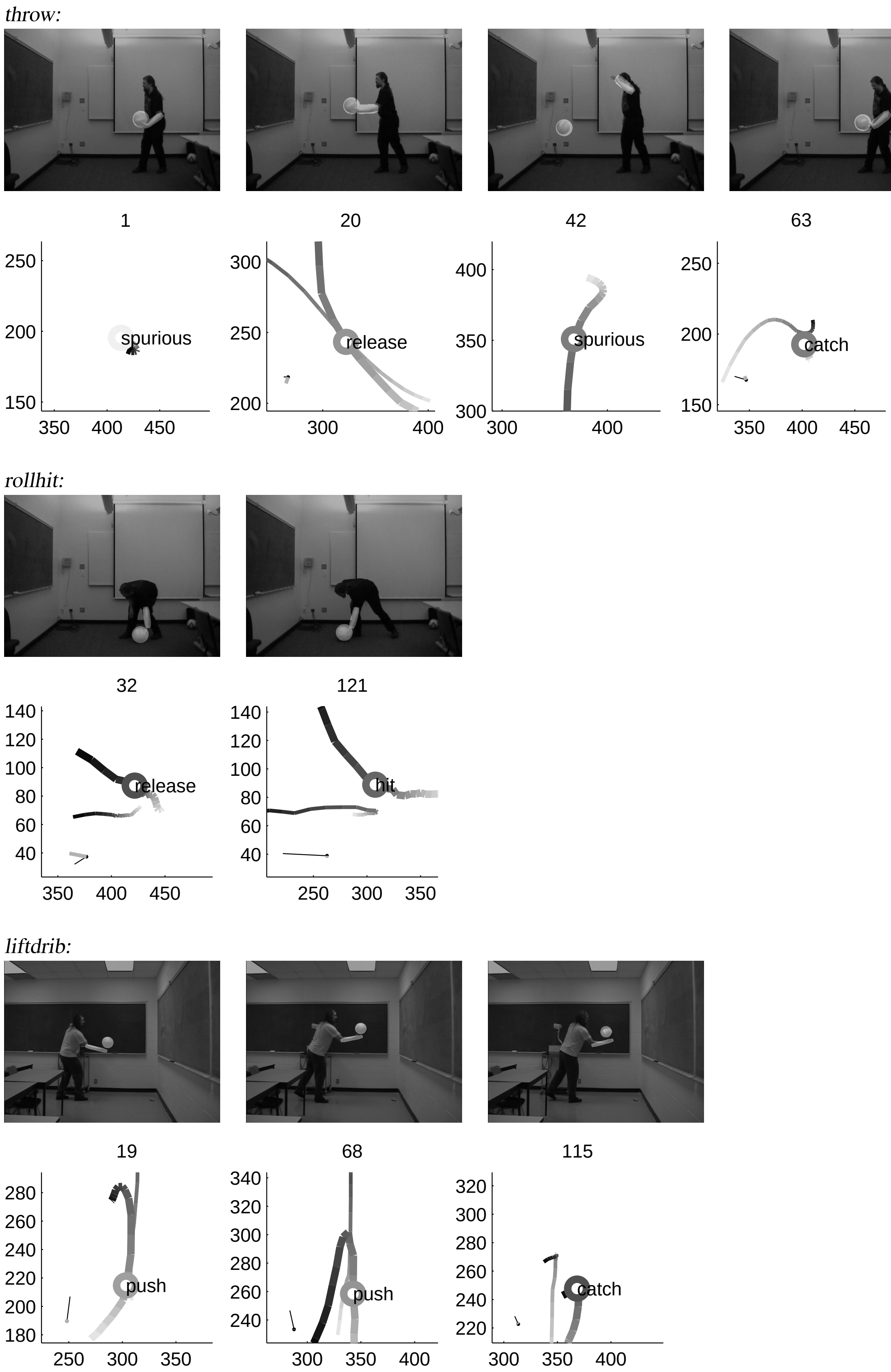

Figure 3. Video and segmentation results for three of the six sequences used in this paper. The ball (circle) and forearm (elongated region) from the tracker are highlighted in each frame. Panels show trajectories of hand (thick line) and ball (thin line) around each event (frame # shown above). Motion discontinuities at events are shown in lower left corner of each panel ($\Delta v =$black, $\Delta a =$grey).

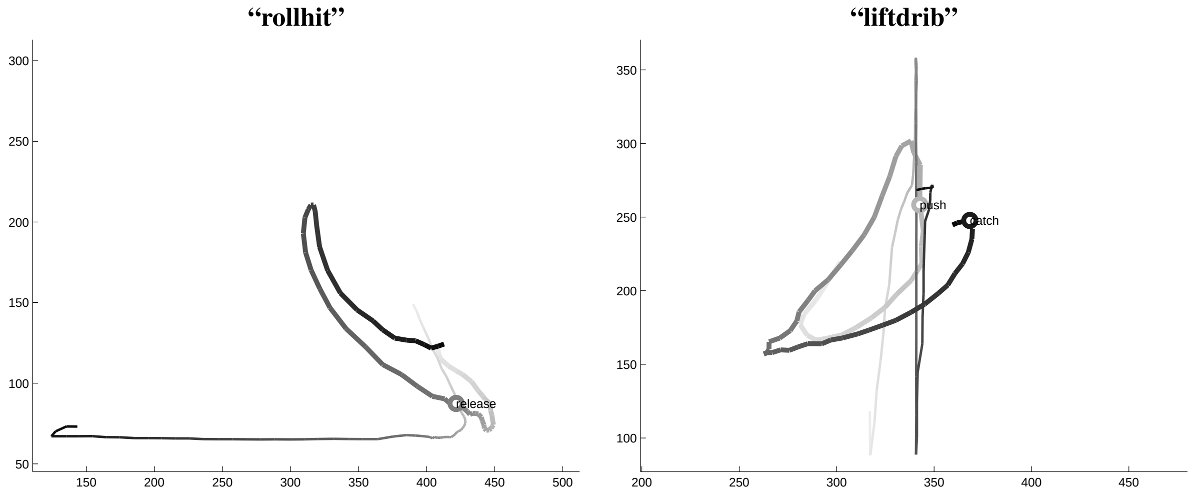

Figure 4. Hand (thick lines) and ball (thin lines) trajectories for motion sequences. Time goes from oldest (grey) to newest (black). Circles show hand segmentation found in Sec. 2.2.

ation, and N is the number of segments in the model. The term, $\lambda > 0$, is the penalty for introducing a segment.

We use a dynamic programming scheme similar to the one described in [5] except that we extend it to fifth order polynomials

$$\begin{aligned}\hat{\mathbf{X}}_n(t;\theta_n) &= \begin{pmatrix} \hat{X}(t) \\ \hat{Y}(t) \end{pmatrix} \\ &= \begin{pmatrix} a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 \end{pmatrix}\end{aligned} \tag{2}$$

The algorithm requires $O(T^2)$ least square fits to complete, where T is the number of frames in the video – the most computationally intensive part of our event detection process. The dynamic programming algorithm outputs a globally optimal hand segmentation over the entire video. We expect it to find the points in time where the hand changes its intentional motion. In practice, this means it gives us a superset of the hand-ball events in the video. Some motion boundaries in the hand do not correspond to hand-ball events, but we are able to automatically discard them as spurious breakpoints.

The quality of the segmentation depends on the value of λ. Fig 5 demonstrates how a segmentation can change over different values of λ. We manually selected robust λ values which provide a stable segmentation. Fig 4 show segmentations for two other sequences. For clarity, only partial trajectories are shown.

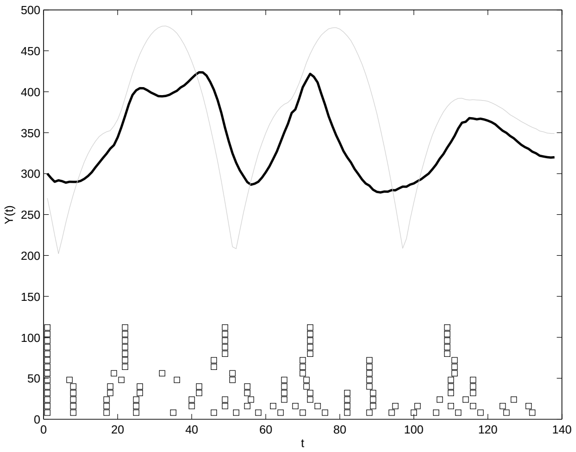

Figure 5. Hand segmentation variation over λ for "liftdrib" sequence. Black and grey curves are $Y(t)$ for the hand and ball respectively. Breakpoints are shown at bottom of plot. Each row shows breakpoints (boxes) for one value of λ. λ varies from zero to 800.

3 Context-dependent gravitational model

The hand motion breakpoints from the dynamic programming segmentation provide candidate points for direct hand-ball events. We use an adaptive ballistic model fitting process on the ball's motion to further analyze these instances. The principle is that the ball's motion can be precisely described using a gravitational (ballistic) motion

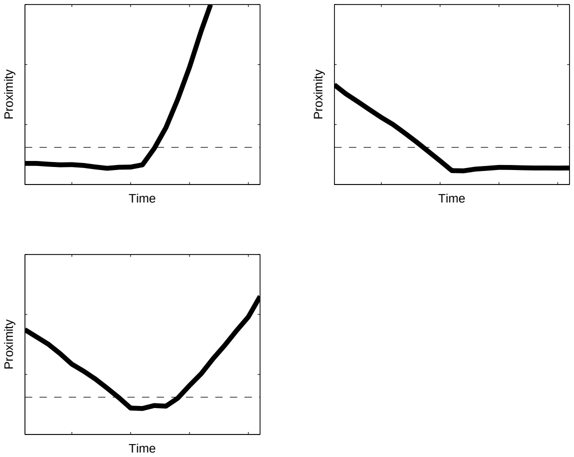

Figure 6. Hand-ball proximity vs. time graphs demonstrating the three interesting classes of contact change at an instant: cessation of contact (top left), onset of contact (top right), and instantaneous contact (bottom left)

model whenever it is not being directly influenced by the hand. A combination of the hand motion model, the hand-ball overlap information from the image, and the ballistic motion of the ball around the breakpoints allows us to filter spurious candidate points and to measure the force impulses and acceleration changes on the ball at the time of the events. We use the hand-ball overlap ontology shown in Fig. 6. Fig. 7 shows a ballistic motion fit around the first breakpoint in the "liftdrib" video. Note the quadratic fits on the left and right side.

3.1. Hand-Ball Overlap Context

We start by creating small time windows with a length of 21 frames around each candidate breakpoint. The breakpoint is at the center of the window and we take the 10 frames after and the 10 frames before it. This window length provides us with enough samples to do local model fitting and analysis around the segmentation boundaries while still representing only 700 ms of video time – short enough to assume that only one instantaneous hand-ball event can occur within the window. This paper adopts the convention from [4] that the segmentation boundary defines two consecutive open time intervals: t_- before, and t_+ after, with t_0 representing the instant.

At each point in time the hand-ball proximity is given from the geometry in the image. We threshold on the proximity to determine the overlap change class of each window. Image overlap can be taken to roughly represent contact between the hand and the ball. The contact change is verified and adjusted later when the ballistic model fitting process is performed. Allowable contact transitions were previously characterized in [4] which gives us a convenient ontology for contact changes within the event window. Five allowable contact change classes were described, but we only find three of them useful for event descriptions:

- $\bar{C}_- C_0 C_+$ – contact onset

- $\bar{C}_- C_0 \bar{C}_+$ – instantaneous contact

- $C_- C_0 \bar{C}_+$ – contact cessation

We observed that this ontology was quite useful because every event window could be clearly described by one unique overlap change class implying a corresponding contact change. In fact, a simple classification scheme that thresholds hand-ball proximity at the beginning, and the end together with the minimum proximity value within a window was always successful in determining overlap change. All event windows which do not demonstrate an interesting contact change (ie, when there was continuous contact throughout or when the hand and ball never contact) are discarded as spurious breakpoints. Only spurious breakpoints which have coincidental image overlap changes can make it past this filter – they are addressed in the ballistic model fitting process.

3.2. Incremental Ballistic Model Fit

Guided by the contact change as context, we perform a sequence of ballistic motion model fits on the ball within the window. We want to fit the ball's motion with a gravitational model precisely when the hand is not contacting the ball. The contact change class provides a rough idea about where the ball may be in free gravitational motion and where it is manipulated by the hand. The simplicity of the distinct contact change classes means that the gravitational motion can only occur in one or two portions of the instantaneous event window: on the t_- (left) side in the case of $\bar{C}_- C_0 C_+$, on the t_+ (right) side in the case of $C_- C_0 \bar{C}_+$ and on the left and right side in the case of $\bar{C}_- C_0 \bar{C}_+$. In particular, the classes indicate whether to use ballistic motion on *on the very first and/or very last frame of the window*. Hence what we may do is incrementally fit quadratic polynomials (the ballistic motion model described in [4]) on the ball starting from the left (and/or right) side of the window. We keep expanding the times of ballistic motion toward the event at the middle of the window until we determine that the ballistic model no longer accurately describes the data.

This style of adaptive fit is successful in giving the precise time when ballistic motion ceases (and/or starts) because the effects of the hand on the motion of the ball are

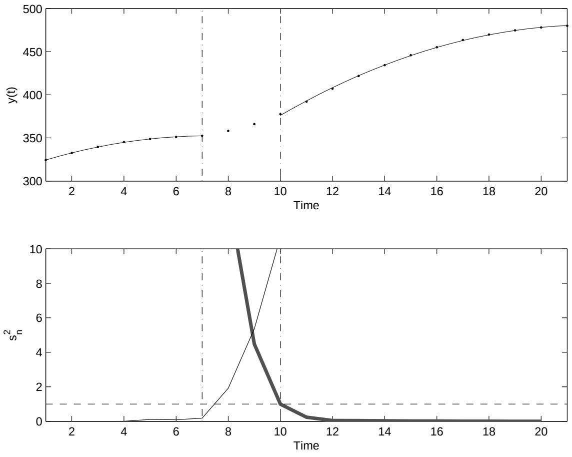

Figure 7. Adaptive fit of gravitational model. (top) Quadratic fit of left and right. (bottom) Error (s_n^2) for gravitational segments. Vertical lines indicate extent of gravational model. For clarity only fit of $Y(t)$ is shown.

much greater than the noise of the tracker. For example, if we perform a least squares quadratic fit on the ball's motion from the left edge of the window until the n^{th} frame we can estimate the error of the fit by the following formula:

$$s_n^2 = \frac{1}{n-1} \sum_{t=1}^{n} \left\| \hat{\mathbf{X}}_n(t) - \mathbf{X}(t) \right\|^2 \qquad (3)$$

where $\mathbf{X}(t) = \left(\begin{array}{c} X(t) \\ Y(t) \end{array} \right)$ is the position of the ball at time t and

$$\hat{\mathbf{X}}_n(t) = \left(\begin{array}{c} \hat{X}(t) \\ \hat{Y}(t) \end{array} \right) = \left(\begin{array}{c} a_0 + a_1 t + a_2 t^2 \\ b_0 + b_1 t + b_2 t^2 \end{array} \right)$$

is a quadratic polynomial representing ballistic motion with coefficients determined by a least squares fit on the ball's position at times 1 through n. (When error values are calculated for the right side of the window, we use equation 3 except t runs from n to the window length.)

So long as the ball follows ballistic motion on frames 1 to n, this average squared error is roughly equal to the tracker noise. As soon as n increases to a point where the hand starts manipulating the ball, we always observe a drastically higher average squared error s_n^2 value. (See the bottom of Fig. 7 for example s_n^2 values as n ranges from 1 to the window length – the thin black plot. The dashed horizontal line represents the expected tracker noise). A threshold on the average residual error provides a good stopping condition on the incremental expansion of the ballistic model. Furthermore, this stopping point is a precise measurement of the onset (or cessation) a hand-ball event. Even though events are reasonably considered instantaneous, some involve a complex physical process that may take several frames within the window. Our fitting method allows us to find the exact start and stop times of such an event. The top of Fig. 7 shows the results of performing an adaptive fit on an event in the "liftdrib" sequence. Note that the fits are done on the left and the right because the overlap context is an instantaneous hit: $\bar{C}_- C_0 \bar{C}_+$.

We can use the event onset and cessation to further filter any spurious sequences. For example, the ball may not have actually come into contact with the hand during the time window. In this case the ballistic motion would be appropriate for the entire window leading our incremental fit to expand the ballistic fit all the way to the other edge of the window. Hence we can filter events as spurious (even though some coincidental hand-ball overlap may have appeared in the image) when the ballistic model clearly describes the motion of the ball during the entire event window. We give ballistic motion fitting preference over the overlap information because our generous image overlap threshold gives some faulty contact change classes when compared to an accurate quadratic motion fit.

The ballistic motion models found here are combined with the event onset (and cessation) times to calculate the force impulses and the acceleration changes that the ball experiences during the event.

3.3. Calculating the Impulses

Using the quadratic polynomial fits on the ball during ballistic motion we can estimate the ball's velocity at the onset (and/or cessation) of the event. We approximate the ball's velocity at all other points using the hand's motion. Recall that the motion of the hand is measured using the fifth order polynomials found in the dynamic programming segmentation. This gives us a well principled estimate of the ball's velocity and acceleration at all points within the event window.

It is a simple matter of subtracting the initial velocity and acceleration vectors of the ball from the final velocity and acceleration vectors to extract the precise instantaneous velocity steps ($\Delta\mathbf{v}$) and acceleration steps ($\Delta\mathbf{a}$). We now infer a hand-ball event caused a force impulse and acceleration change on the ball, and we have estimates for these values. The duration of events and the measured velocity steps provide features we can use to further characterize the events at the non-spurious event breakpoints.

4 Event classification

Given the force impulses as features, can we more precisely classify a hand-ball event? Even in the limited num-

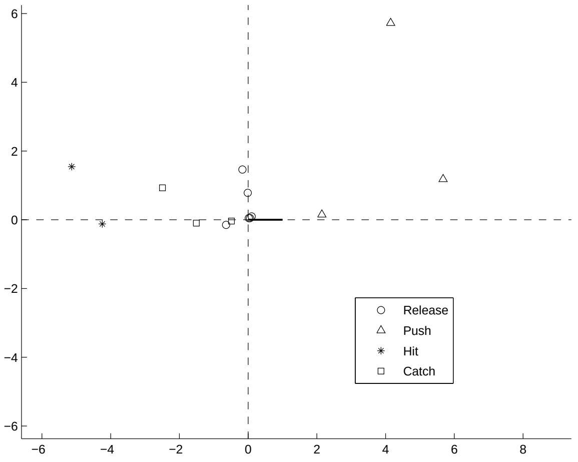

Figure 8. Velocity steps of all events in a special rectangular coordinate frame. Ball's initial velocity is represented as the thick black vector at $(0, 1)$

ber of videos we segmented, the velocity steps we extracted contain some obvious structure that helps answer this question. The events are highlighted in Fig. 8 where a scatter plot of the impulse values is represented using vectors in a rectangular coordinate frame where the initial velocity of the ball at the onset of an event $\mathbf{v}_-$ is represented as $(0, 1)$.

When we manually separate out the different events depending on whether they are considered a release, hit, push, or catch we observe some distinct regualities. These regularities are quite intuitive. For example, we would expect that the velocity step extracted from a catch event would be in the opposite direction to the ball's motion at the instant it is caught and that its magnitude is roughly equal to the magnitude of the ball's velocity because usually the event involves the hand catching the ball and reducing its motion to zero. This structure appears in the data as the squares close to the $(-1, 0)$ point in Fig. 8. We used these categories to provide the criteria for a rough classification scheme for labelling the events in the video sequences. Combining these categories with the contact change we get a simple ontology of hand-ball events and a simple hierarchical decision process to classify the events. This is summarized in Fig. 9

Given more data from additional videos with a larger variety of events, it is reasonable to expect we would see additional regularities. In fact, using the acceleration steps and several other features to create a more complete ontology of hand-ball motion events is an avenue for future work.

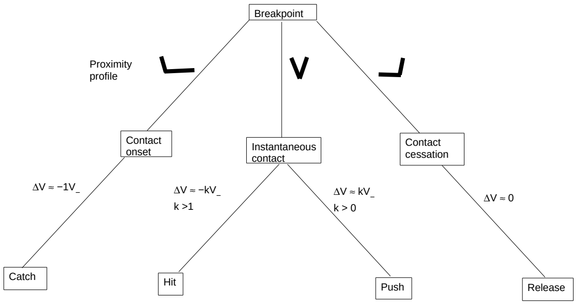

Figure 9. Hierarchical ontology of event motions. Top level distinguishes contact changes (represented by the proximity classes). Next level makes distinctions based on velocity step $\Delta\mathrm{v}$ **relative to the ball's initial velocity** v_-

5 Conclusion

We showed that a piecewise fifth order polynomial segmentation of the hand trajectory was sufficient to find hand-ball interaction in the movies. When combined with proximity and gravitational models, event duration and force impulses may be determined.

There are a number of avenues for future research. An obvious problem is that we track the hand and ball independently, and in a bottom-up fashion. Multiple event models (eg., gravitational and nongravitataional motion) should be incorporated into the tracking process [2]. In addition, we should be able to determine events based on the joint hand-ball motion, incorporating regularities such as that the hand is approaching or following the ball.

Our eventual goal is to use a physics-based model for processing extended motion sequences. Such a system should put additional physical constraints on events, such as energy conservation at collisions, transfer of angular momentum (eg., spin), etc. Furthermore, to represent composite actions, such as dribbling a basketball, we require a representation for extended events. Such events could be described using the event logic proposed in [7].

References

[1] T. Flash and N. Hogan. The co-ordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience*, 5:1688–1703, 1985.

[2] M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *International Conference on Computer Vision (ICCV-98)*, pages 107–112, 1998.

[3] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pat-*

tern Analysis and Machine Intelligence, 25:1296–1311, Oct. 2003.

[4] R. Mann and A. Jepson. Detection and classification of motion boundaries. In *Proceedings of AAAI-2002*, Edmonton, AB, July 2002.

[5] R. Mann, A. Jepson, and T. El-Maraghi. Trajectory segmentation by dynamic programming. In *International Conference on Pattern Recognition (ICPR-02)*, Qubec City, Canada, aug 2002.

[6] R. Mann, A. Jepson, and J. M. Siskind. The computational perception of scene dynamics. *Computer Vision and Image Understanding*, 65(2), Feb. 1997.

[7] J. M. Siskind. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research*, 15:31–90, July 2000.