Trajectory segmentation using dynamic programming

Richard Mann
School of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1 CANADA

Allan D. Jepson Thomas El-Maraghi
Dept. of Computer Science
University of Toronto
Toronto, Ontario M5S 3H5 CANADA

Abstract

We consider the segmentation of a trajectory into piecewise polynomial parts, or possibly other forms. Segmentation is typically formulated as an optimization problem which trades off model fitting error versus the cost of introducing new segments. Heuristics such as split-and-merge are used to find the best segmentation. We show that for ordered data (eg., single curves or trajectories) the global optimum segmentation can be found by dynamic programming. The approach is easily extended to handle different segment types and top down information about segment boundaries, when available. We show segmentation results for video sequences of a basketball undergoing gravitional and non-gravitaional motion.

1 Introduction

We consider the segmentation of a motion trajectory into piecewise polynomial parts, or possibly other forms. Many problems require such a trajectory segmentation, including segmenting 1D data into spline segments [1], segmenting edge chains into lines and/or arcs [6, 5, 8] and processing of piecewise smooth motions, such as cursive handwriting [9].

Segmentation is typically expressed as an optimization problem which trades off model fitting error versus the cost of introducing new segments. When there are multiple segment types, variable costs may assigned so that simpler, lower order, segments are preferred. Alternatively, we can formulate segmentation in a Bayesian framework that assigns a higher prior probability to models containing fewer segments [2], or in a minimum description length (MDL) framework that trades off model complexity for data fit [5]. Since the segmentations are not known in advance, however, these approaches rely on heuristics such as split and merge algorithms [6, 5, 8] or multiscale continuation methods [1]. We show that when the data is described by a single parameterized curve, the global optimum segmentation can be found by dynamic programming. The approach is easily extended to handle different segment types, and top down information about segment boundaries, if available.

Due to the use of dynamic programming, our approach is limited to the segmentation of trajectories having one well defined independent variable. Many problems require low order polynomials to be fit to collections of points or curves in 2D [6, 5, 8]. To apply our approach to such problems, we would require that each grouping hypothesis provides a unique 1D ordering of the data points. This is not a serious issue for the tracking application we consider below, since time provides the required 1D ordering and the individual objects can be tracked unambiguously.

This paper consists of two parts. First we present a novel segmentation scheme which extracts piecewise polynomial segments of a trajectory using dynamic programming. Next we show how this simple algorithm can be applied to motion trajectories of a single object, such as a basketball, undergoing gravitational and non gravitational motion (see Fig. 1). The segmentations we obtain appear to be suitable for the extraction of scene dynamics [7].

2 Trajectory segmentation

Consider the segmentation of a trajectory $\mathbf{X}(t)$ into piecewise polynomial segments. The total segmentation cost is the total sum squared errors in the polynomial fit plus a cost λ for each new segment introduced

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

where $\mathbf{X}(t)$ is the observed motion, $\hat{\mathbf{X}}_n(t;\theta_n)$ is the nth polynomial segment with polynomial coefficients θ_n, and N is the number of segments in the model. The term, $\lambda_n > 0$, is the penalty for introducing segment n. [1]

Minimizing Eq. (1) can be interpreted as maximizing probability of the data according the *penalized likelihood*

[1] Note that Eqn. (1) does not enforce continuity of $\hat{\mathbf{X}}(t)$. This could be done adding constraints to the polynomial coefficients θ_n.

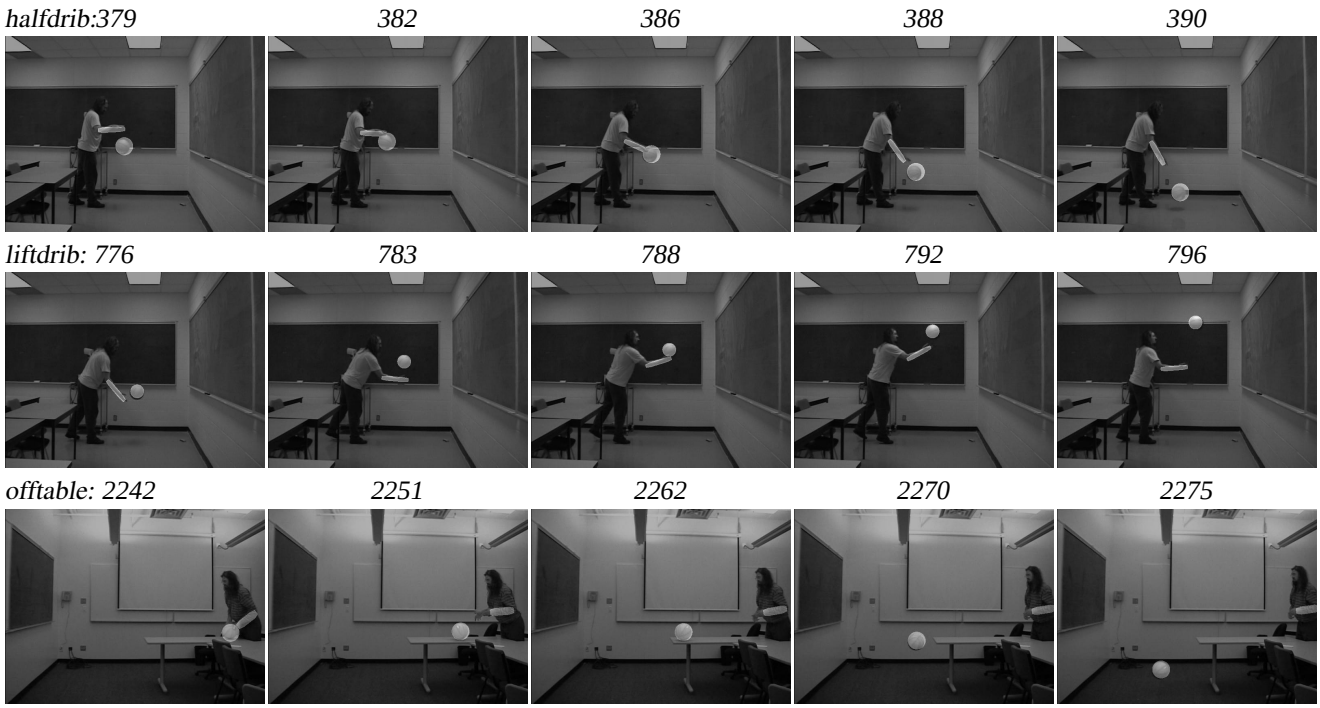

Figure 1. Video sequences. The ball and forearm are highlighted in each frame. See text for details.

function

$$P(\mathcal{X}|\Theta,\Lambda) = \prod_{n=1}^{N} e^{-\lambda_n} \left[\prod_{t=t_{n-1}}^{t_n} \mathcal{N}(\mathbf{X}(t); \hat{\mathbf{X}}_n(t;\theta_n), \sigma) \right] \quad (2)$$

where $\mathcal{X} = \{\mathbf{X}(1), \ldots, \mathbf{X}(T)\}$, $\Theta = \{\theta_1, \ldots, \theta_N\}$, $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$, $\mathcal{N}(x; \mu, \sigma)$ is a normal distribution, and σ is the measurement noise. This is similar to the dynamic programming formulation of stereo matching [3] except that, instead of matching pairs of scan lines, we are searching for an optimal segmentation.

2.1 Dynamic Programming

The global minimum of Eqn. (1) can be found by dynamic programming. Let $S_{t_0}^t$ be the best segmentation up to and including sample t, such that the most recent breakpoint is at $t_0 \in \{1, \ldots, t\}$. At time $t+1$ each segmentation $S_{t_0}^t$ is extended by replacing the cost from t_0 to t with the cost of a new segment from t_0 to $t+1$. S_{t+1}^{t+1} is set to the minimum $S_{t_0}^{t+1}$ over all possible breakpoints $t_0 \in \{1, \ldots, t+1\}$. The algorithm starts with $t = 0$, $S_0^0 = 0$ and increases t from 1 to T, where T is the length of the sequence. The best segmentation is given by S_T^T. At each step the algorithm performs a least squares fit of t polynomial models on the subintervals (t_0, t) for $t_0 \in \{1, \ldots, t\}$. For a sequence of length T, $O(T^2)$ segment fits will be performed.

This algorithm is easily extended to deal with multiple segment types. Suppose there are K different segment types with associated costs λ_k, $1 \leq k \leq K$. Each segment n will have cost $\lambda_n = \lambda_k$ for some k. By assigning smaller costs λ_k to simpler segment types the algorithm trades off data fit for simplicity of the segment type within each fitting interval. If there are K segment types, a total of $O(KT^2)$ segment fits will be performed.

It is often desirable to incorporate top-down information into the segmentation. If a breakpoint is known to occur at a particular time t_0, we perform a restricted search of Eqn. (1) where $t_n = t_0$ for some n. Similarly, if a segment type k is known to occur over interval (t_1, t_2) we constrain $\lambda_n = \lambda_k$, $t_{n-1} = t_1$, and $t_n = t_2$ for some segment n.

3 Segmentation of image motion

We consider the segmentation of the motion trajectory of an object, such as a basketball, undergoing gravitational and nongravitational motion (see Fig. 1). The ball may fall, bounce, or roll along a horizontal surface. In addition, an active object, such as the hand, may exert forces on the ball by pushing, lifting, or holding. In each sequence the forearm and the ball were tracked by an adaptive view-based tracker described in [4].

Provided the depth variation is small relative to the absolute scene depth (ie., a weak perspective model), we can model the projected motion of the ball in the image by quadratic motion segments:

$$\hat{\mathbf{X}}(t) = \begin{pmatrix} \hat{X}(t) \\ \hat{Y}(t) \end{pmatrix} = \begin{pmatrix} a_0 + a_1 t + a_2 t^2 \\ b_0 + b_1 t + b_2 t^2 \end{pmatrix} \quad (3)$$

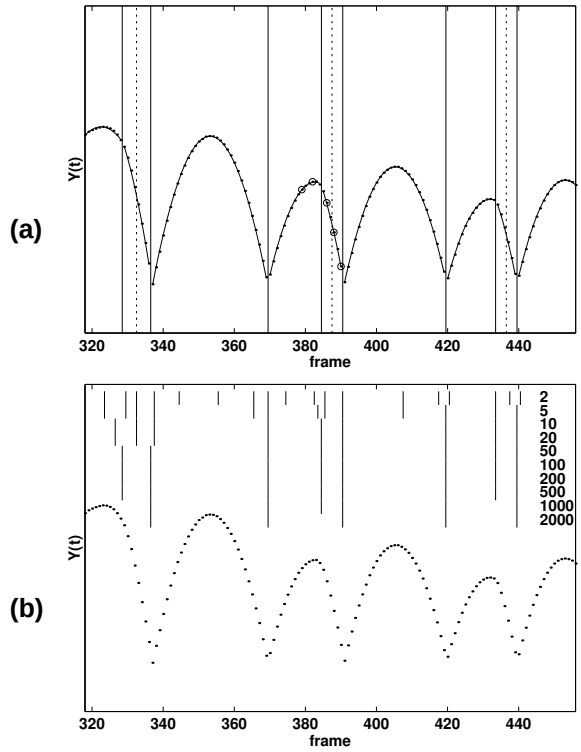

Figure 2. (a) Segmentation of *halfdrib* into quadratic pieces for $\lambda = 100$ (circles denote frames shown in Fig .1, dotted lines denote missed breakpoints). (b) Stability of segmentation as λ varies.

This constant acceleration model is appropriate both gravitational or nongravitational motion, provided the hand exerts a roughly constant force on the ball. Special cases of this motion include ballistic (gravitational) motion:

$$\begin{pmatrix} \hat{X}(t) \\ \hat{Y}(t) \end{pmatrix} = \mathbf{P} + h(t)\mathbf{D}(\theta) + v(t)\begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (4)$$

where $\mathbf{P} = (P_x, P_y)^T$ is the starting point, $h(t) = h_1 t$ is the translation speed, $\mathbf{D}(\theta) = (\cos\theta, \sin\theta)^T$ is the direction of translational motion, and $v(t) = v_1 t + v_2 t^2$ is the gravitational motion. The acceleration due to gravity is $\mathbf{g} = (0, 2v_2)^T$ pixels/frame^2. A second case is rolling motion:

$$\begin{pmatrix} \hat{X}(t) \\ \hat{Y}(t) \end{pmatrix} = \mathbf{P} + h(t)\mathbf{D}(\theta) \quad (5)$$

where $h(t) = h_1 t + h_2 t^2$. h_2 has nonzero values for deceleration due to (sliding) friction, while h_1 and h_2 are both zero for a resting object.

In this paper we consider only fronto parallel (side view) motion thus the image motion becomes $(\hat{X}(t), \hat{Y}(t)) = (a_0 + a_1 t, b_0 + b_1 t + b_2 t^2)$ for the gravitational model and $(\hat{X}(t), \hat{Y}(t)) = (a_0 + a_1 t + a_2 t^2, b_0)$ for the rolling model.

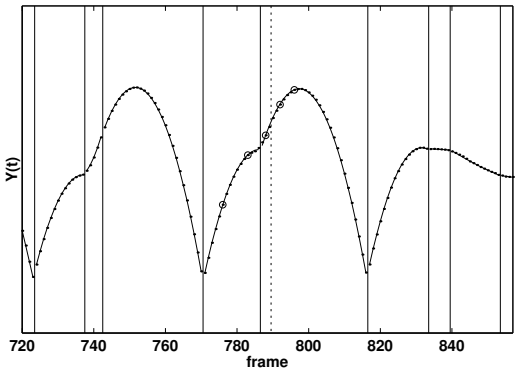

Figure 3. Segmentation of *liftdrib* into quadratic pieces (dotted line denotes a missed breakpoint).

Fig. 2a shows the segmentation of the *halfdrib* sequence into piecewise quadratic segments. Note that every interval is fit by separate polynomials, $\hat{X}(t)$ and $\hat{Y}(t)$, but for cases where the motion is essentially vertical, only $\hat{Y}(t)$ is shown. The bounces (frames 337–369, 391–439) and collisions (frames 337, 370, 391, etc.) are easily detected. From the first bounce we estimate gravity at approximately 1.67 pixels/frame^2 and the tracker noise σ at approximately 0.59 pixels. The system also finds the onset of pushing on *every other* bounce (frames 330, 385, and 435). During pushing, the acceleration is well modeled by a constant acceleration of approximately 2.1 to 2.2 pixels/frame^2. Note that while the segmentation is stable over a wide range of λ (Fig. 2b), we are unable to detect the *removal* of the hand. From the video, we know that the hand was removed (and the ball returned to gravitational motion) sometime after the onset of pushing, but before the ball hit the ground (see the dotted lines in Fig. 2a).

Fig. 3 shows the segmentation results for the *liftdrib* sequence. Again, the segmentation is imperfect: The first lift is detected (frames 738–742), with an acceleration of approximately -2.5 pixels/frame^2 (ie., upwards), but the removal of the hand after the second lift (frame 789) was missed. At the end of the sequence the hand is holding the ball. Here the motion is not well modeled by quadratic segments, and over segmentation results. In Sec. 3.2 we use the hand's motion to improve the segmentation for these two sequences, but first we demonstrate segmentation into multiple motion types.

3.1 Multiple motion types

Fig. 4a shows the tracking data (both $X(t)$ and $Y(t)$) for the *offtable* sequence. Here the ball rolls along the table, falls, bounces on the ground, hits the wall, and continues to bounce on the ground. Fig. 4b shows the segmentation into both quadratic and linear models. To enforce a prefer-

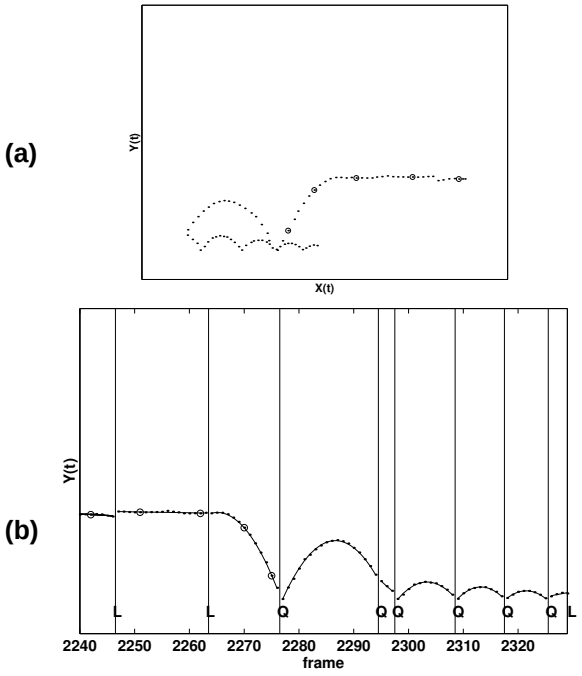

Figure 4. Segmentation of *offtable*. (a) Motion trajectory. (b) Quadratic (Q) and linear (L) segments.

ence for simpler linear models, we used $\lambda_L = 50$ for linear models. ($\lambda_Q = 100$ for quadratic models, as before.) A linear model is fit while the ball is rolling on the table, while a quadratic model is fit during falling and bouncing. Note that we (correctly) detect an extra breakpoint in $X(t)$ (frame 2295) where the ball bounces off the wall. Also note that the motion ends with a rolling segment once the bounces become small. The extra breakpoint (frame 2246) is caused by a tracker error.

3.2 Exploiting context: hand proximity

When the hand and the ball overlap in the image, it is likely that the hand is actually touching the ball in the scene. During such contact, the hand may apply arbitrary forces to the ball, hence a quadratic motion model is inappropriate.

To handle such cases, we introduce a third segment type (H, for "hand") which allows arbitrary piecewise linear motion during an interval. We set the segment cost $\lambda_H < \min(\lambda_L, \lambda_Q)$ to ensure that only the hand model will be fit during contact intervals.

Fig. 5 shows the segmentation of *liftdrib* using the hand model. The bars at the bottom of the figure show intervals where the hand and the ball contact in the image. The hand is contacting the ball during both lifting segments, and at the end of the sequence, where the hand is holding the ball. Note that there are brief intervals of apparent contact (frames 761, 821) where the the ball moves directly behind the hand. Such apparent contact does not alter the segmentation as long as the noise within the contact interval is smaller than the cost of introducing a new hand segment.

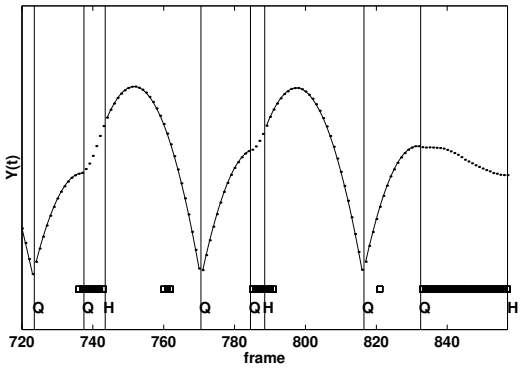

Figure 5. Segmentation of *liftdrib* using hand proximity (H denotes hand segments).

Since we cannot determine the proximity of the hand and the ball exactly, we use a rather loose tolerance. This finds contacts well, but tends to overestimate the duration of contact. However, since there is no fitting cost associated with hand segments, we need a way to prevent the hand segments from absorbing their entire contact intervals. To achieve this, we add a duration cost, $\hat{\sigma}T$, to each hand segment, where T is the length of the interval and $\hat{\sigma}$ an estimate of the tracker noise. Fig. 5 shows the segmentation of *liftdrib* for $\lambda_H = 20$ and $\hat{\sigma} = 0.5$. The addition of the hand model with duration cost yields a very good segmentation (compare with Fig. 3).

References

[1] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[2] K. Bubna and C. Stewart. Model selection and surface merging in reconstruction algorithms. In *ICCV-98*, pages 895–902, 1998.

[3] I. Cox. A maximum likelihood n-camera stereo algorithm. In *CVPR-94*, pages 733–739, 1994.

[4] T. F. El-Maraghi. *Robust Online Appearance Models for Visual Tracking*. PhD thesis, Department of Computer Science, University of Toronto, in preparation.

[5] M. Li. Minimum description length based 2-d shape description. In *ICCV-93*, pages 512–517, 1993.

[6] D. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, Norwell, MA, 1985.

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

[8] P. Rosin and G. West. Nonparametric segmentation of curves into various representations. *IEEE Trans. Pattern Analysis and Machine Intell.*, 17(12):1140–1153, December 1995.

[9] Y. Singer and N. Tishby. Dynamical encoding of cursive handwriting. *Biological Cybernetics*, 71:227–237, 1994.