Multidimensional Motion Segmentation and Identification

Chunmei Lu Haizhu Liu Nicola J. Ferrier*

Robotics Laboratory, Dept. of Mechanical Engineering
University of Wisconsin-Madison, Madison, WI 53706-1572
E-mail: ferrier@engr.wisc.edu

Abstract

Accurate tracking can facilitate the automatic extraction of metric information from video analysis. Many tracking systems rely on a sufficiently accurate dynamic model. These dynamic models must be either known a priori or learnt. This paper addresses the problem of determining dynamical system models from observed visual motion where it is assumed that the motion cannot be modeled by a single dynamical system. The changes in motion (from one system to another) need to be detected. Previous work has dealt with maintaining multiple hypotheses. For repetitive motion, rather than maintain multiple hypotheses one can learn the dynamic models that apply and identify the changes between the models. Specifically, a method for high dimensional motion segmentation is presented. By using a two-step recursive least square algorithm, break points of system dynamics, at which a model switching must be performed, are predicted. After segmentation, system identification techniques can be used to fit dynamic models.

1. Introduction

Our research addresses the problem of measuring repetitive motion from video sequences. Assuming that motion can be described by a sequence of simple dynamic models, the goal of this paper is to present a method to automatically segment complex motion into a sequence of simple models where no *a priori* assumptions are made about the number of simple models that comprise the full motion. The application under consideration is the automatic measurement of human motion while performing repetitive tasks. Different tasks will be described using sequences of simple linear dynamic models. Because motions are typically described in high dimensions (six or more parameters), we present a multi-dimensional segmentation algorithm based on univariate active forgetting segmentation methods [1].

*This work was supported by NSF (IRI-9703352) and Univ. of Michigan (as a subcontract for DHHS/PHS/CDC).

1.1. Previous work

Measurement of human motion from video sequences has been approached using numerous techniques (due to space constraints, here we mention only a very small fraction of the research effort in this area). Early work used synthetic images and constraint satisfaction methods [14]. Hogg [10] used structural/kinematic models of human limbs and extensive search to locate humans in real image sequences. Recent work often incorporates structural models (e.g. [15], or see [17] for a survey). Much of the work on gesture recognition and human motion tracking (e.g. [7, 9, 20, 21]) does not require highly accurate reconstruction of the motion in three dimensional space. Qualitative results are often sufficient for the applications of these systems.

For some applications, such as video analysis for animation, motion must be recognized or reconstructed in three dimensions with sufficiently high accuracy. Many systems present an analysis of human motion where the data often comes from specialized hardware (e.g. commercially available LED tracking systems or magnetic tracking systems) which attain highly accurate motion tracks.

Other work emphasizes analysis, not extraction, of motion. For example [6, 16] describe higher level representations, with emphasis on motion analysis strategies or recognition and not the extraction of motion data from video sequences.

In this paper we concentrate on methods to facilitate the low-level extraction of motion from video sequences. We present a method to segment the motion and fit models to the individual segments. This information can then be used within a dynamic model based tracking system to improve tracking performance.

1.2. Multi-model approaches

While a human is performing a task the limbs (upper arm, lower arm, hand) are in motion. For *portions* of this motion, the movement of the individual limbs can be de-

scribed by a linear dynamical system [4]. For complex motions describing the *entire* task, the motion can be represented as a sequence of simple movements [5].

Previous efforts have sought to automatically determine which dynamic model is applicable. Bregler [5] used a sequence of linear dynamic models to describe leg motion. Individual models were used for phases of gait such as the swing phase and the support phase. The state space of these dynamical systems are the velocity screw vector describing the motion of the limb. A cyclic hidden Markov model (HMM) is used to transition between applicable dynamic models. Similarly, Isard and Blake [11] use a multi-hypothesis approach with the CONDENSATION tracking algorithm to identify which dynamic model, from a fixed set of models, to use for motion. Torr [18] also uses a multi-hypothesis approach to switch between motion models for structure-from-motion. The number of states for the HMM or for the multi-hypothesis frameworks was assumed to be *known* and the emphasis of these works was in the determination of which model to apply during a particular sequence.

In contrast, this paper explores the automatic determination of motion states from image data. Because our long term goal is to be able to automatically analyze video data of humans performing various tasks, no assumptions about the number of states can be made. We present a method to segment motion data and fit dynamic models to the individual segments. A recurrent network architecture has been proposed [19] to simultaneously estimate motion and segment the scene. This segmentation, however, is a spatial segmentation to determine segment motion from multiple moving objects in a scene. Here we deal with temporal segmentation.

2. Visual motion data

For the purposes of this paper we are using motion data extracted using an implementation of a contour-based tracking system. To verify the segmentation we consider affine motion models (explained below) and track simple objects moved by a robot arm (in order to control when a change in model occurs thus providing ground truth).

Contour based tracking has been successfully employed in tracking non-rigid objects, such as lips, hands and faces. Most contour based tracking systems detect high contrast edges [2] or image motion[8]. The object to be tracked is represented by a parametric spline curve called a deformable template or snake [3]. The control points of this curve completely determine the snake. By allowing arbitrary freedom of the control points, the snake can act in an unstable way. To avoid this, the shape of the target outline in some particular position, if it is known, can act as a *template* that the snakes tend to relax on to achieve stable behavior[2]. The shape template can also reduce the state space used to describe the state[8]. For example, assuming perspective projection and a planar object undergoing rigid motion in space, the image motion of the object is approximately *affine* and only six parameters are required to describe the motion. Rather than specifying the motion of the control points (a state space that may be arbitrarily large for some shapes), and, provided perspective effects are not too strong, a good approximation to the curve shape as it changes over time can be obtained by specifying the six affine components.

2.1. Dynamical models

The motion of the contour can be described by formulating a dynamical system. The *state* of the system is comprised of the affine components. The motion of the curve is described by the change of those affine components.

Motion tracking is based on a combination of *prediction* and *measurement*, which needs a statistical framework to combine deterministic and stochastic processes. An autoregressive model is an appropriate model to describe motion in one frame based on motion in the previous frames, because the essence of the AR model is using previous data to predict future data. We only consider the second order AR models since the first-order dynamical model can not adequately describe motions of interest, such as motion of human hands[3].

Let $\mathbf{X}(t_k)$ denote the 6-dimensional affine component vector at time t_k. The second-order AR process has the form

$$\mathbf{X}_{t_k} - \bar{\mathbf{X}} = \mathrm{A}_2(\mathbf{X}_{t_{k-2}} - \bar{\mathbf{X}}) + \mathrm{A}_1(\mathbf{X}_{t_{k-1}} - \bar{\mathbf{X}}) + B_0\mathbf{e}_k \quad (1)$$

where A_2, A_1 and B_0 are all 6×6 matrices. $\mathbf{e}_k$ is the Gaussian noise with zero mean and variance σ_k^2 and $\bar{\mathbf{X}}$ is the mean. Equation 1 can be expressed more compactly by defining a *state vector* $\mathcal{X}(\mathrm{t_k}) = \left(\mathbf{X}_{t_{k-1}}, \mathbf{X}_{t_k}\right)^T$ and then writing

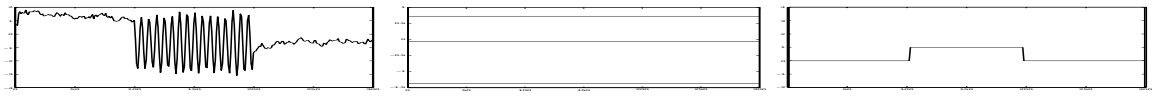
$$\mathcal{X}(\mathrm{t_k}) - \bar{\mathcal{X}} = A(\mathcal{X}(\mathrm{t_{k-1}}) - \bar{\mathcal{X}}) + B_0\mathbf{e}_k \quad (2)$$

where

$$\bar{\mathcal{X}} = \begin{pmatrix} \bar{\mathbf{X}} \\ \bar{\mathbf{X}} \end{pmatrix}, \; A = \begin{pmatrix} 0 & I \\ A_2 & A_1 \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 \\ B_0 \end{pmatrix}.$$

The system matrix of the process, A, fully describes the deterministic part of the model, and matrix B describes the stochastic part of the model.

Motion can be described by a dynamical system model such as the AR model. However, as is evident in these models, parameters (A_1, A_2, B_0) are required.



(a) (b) (c)

Figure 1. (a) Motion, (b) Segmentation by Andersson's algorithm fails: each line shows an AR parameter value, the constant values indicate no segmentation was found. (c) RLS segmentation by model-fitting error correctly segments the motion (a step indicates the location of a segmentation).

3. Recursive model identification

For motion of simple planar objects, we consider the six affine parameters of motion. During a constant motion (single dynamical system), the dynamical model should remain fixed. However, we conjecture that changes in these affine parameters indicate changes within the underlying dynamical system. For example, Fig. 1(a) shows obvious transition in motion, characterized by one oscillating motion in the middle of two stationary states. There will be roughly three underlying dynamic models.

For our model above, the problem is to detect the change of the dynamical parameters A(A_1 and A_2) over time. A basic scheme for this problem is *recursive parameter estimation*. With the term *recursive* we mean that the new value of the estimated parameters is equal to the previous value plus a *correcting term* which will depend on the most recent measurements. A typical description of a time-discrete (1-dimensional) system with jumping parameters is the state space model

$$\left.\begin{array}{c} \theta(t+1) = \theta(t) + w(t) \\ y(t) = \varphi^T(t)\theta(t) + e(t) \end{array}\right\} \quad (3)$$

where $\theta(t)$ is an n-dimensional vector containing the true parameters describing the system at time t, $\varphi(t)$ is a vector containing the old inputs and outputs, and $e(t)$ and $w(t)$ are disturbances with variances $R_2(t)$ and $R_1(t)$ respectively.

A possible estimation of θ is given by the well known Kalman filter:

$$\left.\begin{array}{c} \hat{\theta}(t) = \hat{\theta}(t-1) + K(t)(y(t) - \varphi^T(t)\hat{\theta}(t-1)) \\ K(t) = \frac{P(t-1)\varphi(t)}{R_2(t)+\varphi^T(t)P(t-1)\varphi(t)} \\ P(t) = P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{R_2(t)+\varphi^T(t)P(t-1)\varphi(t)} + R_1(t) \end{array}\right\} \quad (4)$$

In fact, if the disturbance $w(t)$ in eq. (3) is Gaussian with a known variance $R_1(t)$, the above Kalman filter gives the best estimate of θ[13].

Andersson[1] presented a segmentation approach based on finite-Gaussian sum approximation. His data segmenting algorithm is based on several parallel models. Each model is recursively estimated by an algorithm of Kalman filter type and is updated independently.

A common problem with both of the above algorithms is that they require some priori knowledge of the system to be identified, i.e., $R_1(t)$ for Kalman filter and $R_2(t)$ (the variance of $e(t)$) for Andersson's method. Without this information, both of the above algorithms are likely to exhibit over- or under-segmentation. Fig.1(b) gives an example demonstrating that even with an estimation of R_2 using Andersson's algorithm, the segmentation still fails. Estimation of parameters R_1 or R_2 is difficult and hence generally these values are attained by trial and error. Considering the high dimension case of interest here, in which we need to monitor 72 parameters in two 6x6 matrices, it is impossible to test and try all the possibilities. A more robust and efficient high dimensional segmentation method is needed.

4. Two-step recursive least square segmentation

Using $W(t)$ to model the changing parameters, the affine model we considered above can be described as:

$$\left.\begin{array}{c} \mathcal{A}(t+1) = \mathcal{A}(t) + W(t) \\ Y(t) = \mathcal{A}^T(t)Y(t-1) + E(t) \end{array}\right\} \quad (5)$$

$$\text{where} \quad Y(t) = \mathcal{X}(\mathrm{t_k}) - \bar{\mathcal{X}} \ \text{ and } \ A^T = \begin{pmatrix} 0 & A_2 \\ I & A_1 \end{pmatrix}$$

Minimizing the criterion function

$$V_t(\mathcal{A}) = \sum_{k=1}^{t} \beta(t,k)\|Y(k) - \mathcal{A}^T Y(k-1)\|^2$$

$$\text{with} \quad \beta(t,k) = \prod_{j=k+1}^{t} \lambda(j)$$

we obtained the following high dimensional recursive least square (HRLS) algorithm:

$$\left.\begin{array}{c} \hat{\mathcal{A}}(t) = \hat{\mathcal{A}}(t-1) + L(t)(Y(t) - \hat{\mathcal{A}}(t-1)^T Y(t-1))^T \\ L(t) = \frac{P(t-1)Y(t-1)}{\lambda(t)+Y^T(t-1)P(t-1)Y(t-1)} \\ P(t) = \frac{1}{\lambda(t)}\{P(t-1) - \frac{P(t-1)Y(t-1)Y^T(t-1)P(t-1)}{\lambda(t)+Y^T(t-1)P(t-1)Y(t-1)}\} \end{array}\right\} \quad (6)$$

where $P(t)$ is *adaptation gain*, and $\lambda(t)$ is called the *forgetting factor*, which is used to discount old measurements. λ is chosen as a constant 0.98 in our experiments.

Our high dimensional motion segmentation algorithm is partially based on the above HRLS algorithm. Essentially it consists of a two-step RLS algorithm:

1. HRLS is applied to the high dimensional affine model system and gives the prediction of affine parameter matrix $\mathcal{A}(\mathrm{t})$. The changes of our multivariable system are be represented by the difference matrix of adjacent parameter matrices.
2. The MSE of the difference matrix is computed and an additional one-dimensional RLS is applied to this one-dimensional system to detect the final segmentation points.

Unlike the previous system-identification-based segmentation methods, which makes the segmentation completely based on the output of the system identification algorithms, we determine the segmentation step by step according to the model-fitting error. By doing this, we avoid the awkward problem that the performance of the algorithm depends on how much prior knowledge we have about the system.

To summarize, the high dimensional segmentation algorithm is given as follows (Fig. 2 shows the result of each step):

1. The data is filtered to reduce noise and select *useful* portions of the original data. A low-pass filter is applied to smooth the data and the mean value of the data is then subtracted to make the model more compact.
2. Apply HRLS to the affine data and compute the difference matrix of the adjacent parameter matrices. This difference matrix gives a reasonable measurement of changes of our affine parameters.
3. The norm of this difference matrix is computed at each time unit, converting our multivariable system to a one-dimensional time series. Several kinds of matrix norms were tried and we found their performance to be similar with respect to the segmentation achieved. The widely used MSE (mean square error) was adopted in our experiments. For $A = (a_{ij})$, this norm is defined as
$$\|A\| = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2.$$
4. Apply one dimensional RLS to the MSE of the difference matrix. The parameter of this time series is estimated.
5. Determine peak points of the resulting parameters of step 4. A high pass filter is applied to determine the most promising spike points.
6. Perform model fitting to each segmented interval as described in section 5. If the fitting error is great than a threshold, go back to step 5 for a finer segmentation until the fitting threshold is satisfied.
7. A merge technique[12] might be adopted to combine the data from adjacent segments which agree on the system parameters. This could be useful to avoid oversegmentation in the case of very noisy data.

The above criteria for determining the segmentation can also be applied to the one-dimensional case. We found that without any *a priori* information about error variance (R_2), one-dimensional RLS gives a good segmentation (see Fig. 1(c)). Andersson's method requires an estimate of the error variance (R_2), an incorrect estimate yields poor segmentation.

5. Model fitting

Given the segmentation points, a model fitting must be performed at each interval. The problem is to determine the parameters matrix A(A_1 A_2), B_0 and $\bar{\mathbf{X}}$ in our second order AR model. Given the data of affine components $\mathbf{X}_1, \ldots, \mathbf{X}_M$ from an image sequence, the learning algorithm is based on maximizing the log-likelihood function

$$L(\mathbf{X}_1, \ldots, \mathbf{X}_M | A_1, A_2, C, \bar{\mathbf{X}}) = -(M-2)\log\det B_0 - \frac{1}{2}\sum_{k=3}^{M}|B_0^{-1}(\mathbf{X}'_k - A_2\mathbf{X}'_{k-2} - A_1\mathbf{X}'_{k-1})|^2 \quad (7)$$

$$\text{where} \qquad \mathbf{X}'_k = \mathbf{X}_k - \bar{\mathbf{X}} \text{ and } C = B_0 B_0^T \quad (8)$$

which leads to the following algorithm presented in[3]

1. First, compute R_i, $i = 0, 1, 2$ and auto-correlation coefficients R_{ij} and R'_{ij}, $i, j = 0, 1, 2$ from affine data sequence $\mathbf{X}_1, \ldots, \mathbf{X}_M$:
$$R_i = \sum_{k=3}^{M}\mathbf{X}_{k-i},\ R_{ij} = \sum_{k=3}^{M}\mathbf{X}_{k-i}\mathbf{X}_{k-j}^T, \quad (9)$$
$$R'_{ij} = R_{ij} - \frac{1}{M-2}R_i R_j^T \quad (10)$$
2. Parameter estimates $\hat{A}_1, \hat{A}_2$ and $\hat{\mathbf{D}}$ are given by
$$\hat{A}_2 = (R'_{02} - R'_{01}R'^{-1}_{11}R'_{12})(R'_{22} - R'_{21}R'^{-1}_{11}R'_{12})^{-1} \quad (11)$$
$$\hat{A}_1 = (R'_{01} - \hat{A}_2 R'_{21})R'^{-1}_{11} \quad (12)$$
$$\hat{\mathbf{D}} = \frac{1}{M-2}(R_0 - \hat{A}_2 R_2 - \hat{A}_1 R_1) \quad (13)$$

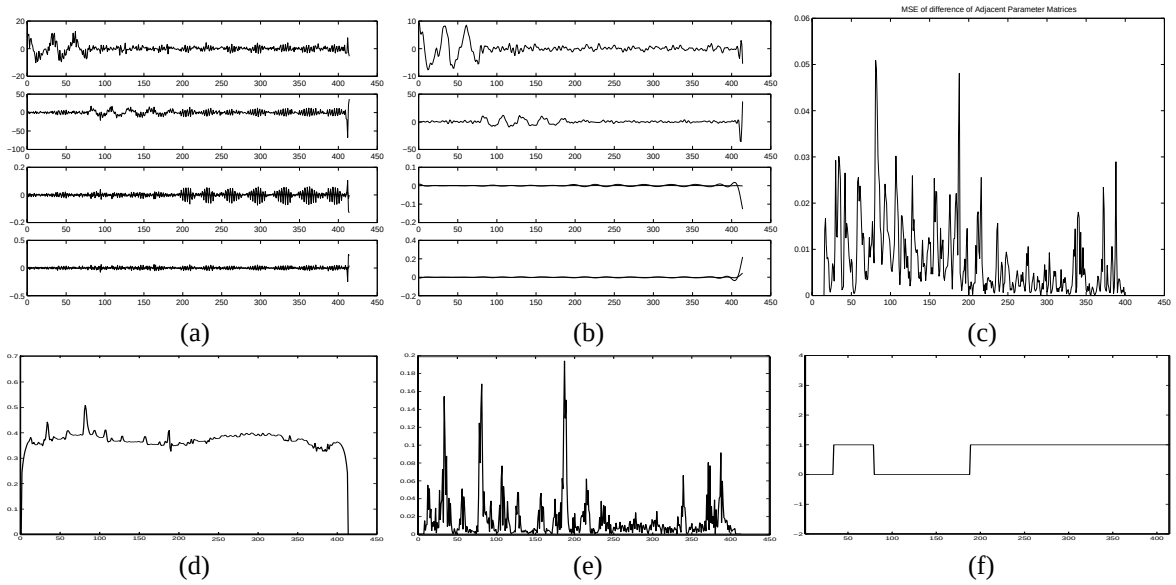


Figure 2. The steps for segmentation: (a) Original affine data measured from observation of a human moving an object. The plots from top to bottom are X-translation, Y-translation, divergence and curl, deformations, respectively; (b) Low pass filtered motion data; (c) MSE of difference matrix; (d) Parameter prediction of MSE of difference matrix; (e) Apply high-pass filter to determine the peak points; and (f) Segmentation result — a step change, up or down, indicates the location of a computed segmentation point.

3. The covariance coefficient B_0 is estimated as a matrix sequence root $\hat{B}_0 = \sqrt{\hat{C}}$ where

$$\hat{C} = \frac{1}{M-2}(R_{00} - \hat{A}_2 R_{20} - \hat{A}_1 R_{10} - \hat{\mathbf{D}} R_0^T) \quad (14)$$

The learned model is then simulated by generating a pseudo-random Gaussian noise and driving the discrete dynamics with the system matrix.

6. Experimental results

Example segmentation and model fitting results are shown in Fig. 2-6. Fig. 2 shows the steps of the segmentation process. The motion was generated by tracking a hand held tool. Figures 3 gives the dynamic model predictions of each affine component after model fitting. Because a human generated the motion, it is not periodic, however the individual cycles of motion are correctly identified. Precise cyclic motions were generate by tracking an object moved by a robot. Fig. 4 shows an example configuration. The camera observes repetitive motion of the object. Figures 5 and 6 show the segmentation obtained for the robot generated motion. The two rightmost columns of 6 gives the predictions of each affine component after model fitting. In all the examples shown, after the first segmentation, model fitting is performed and intervals in which the fitting error is great than a threshold are re-segmented. For the robot generated motion, the final segmentations are agree with the commanded motion, while for the human generated motion, the final segmentations agree qualitatively with the motions performed. Segmentation using synthetic data, whose ground truth was known, was performed for various SNR (graphical results are not shown). We found that for small SNR ($SNR \approx 2$, $R_2 = 0.5$) segmentation fails. Hence segmentation requires reasonable tracking data.

7. Summary

We have presented a method to automatically segment high dimensional motion data such that each individual segment can be described by a simple linear model. No *a priori* knowledge about the number of possible segments (or the dynamic models applied on the segments) were required. However, the thresholds to determine the accuracy of the model fit (and hence the segmentation) must be chosen. Experimental results show that the method works on motions tracked using a contour-based tracker with an affine motion model. The extension to any high dimensional model is straightforward.

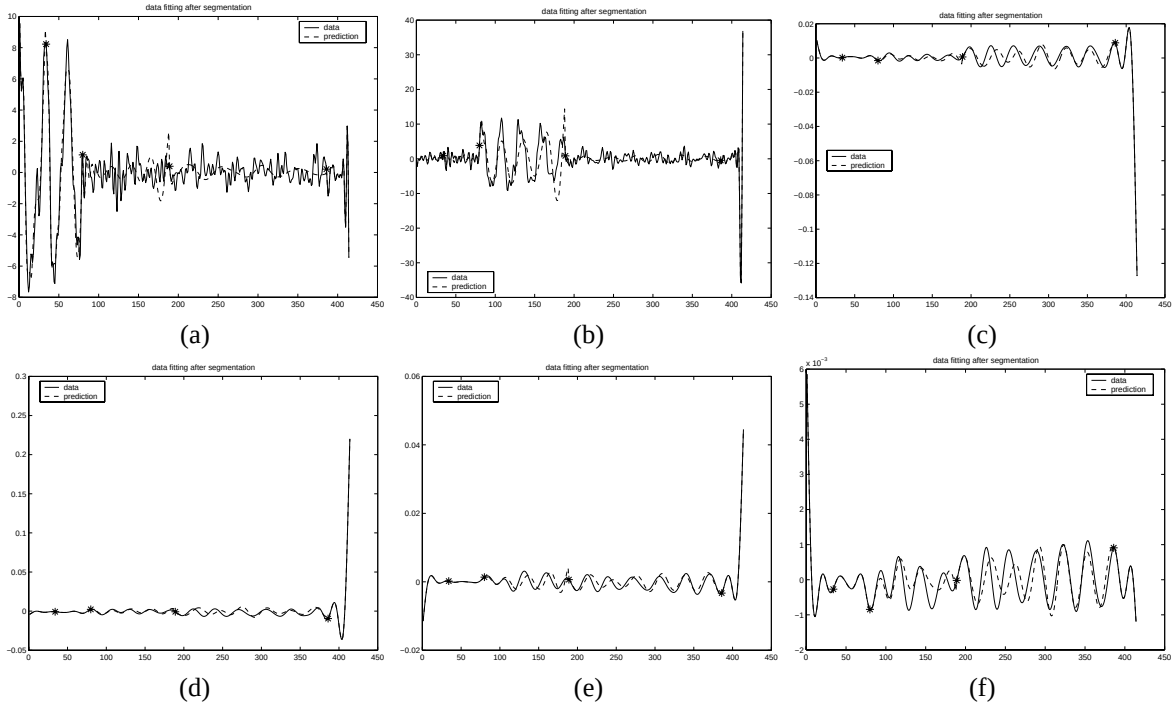


Figure 3. Dynamic model fitting after segmentation (using motion from figure 2 with the computed segmentation points shown by astericks, '*'). The plots compare the the prediction from learned model (line with '-.') and original measurement(solid line) for the six affine components: (a) horizontal translation, (b) vertical translation, (c) divergence, (d) deformation, (e) deformation, (f) curl

References

[1] P. Andersson. Adaptive forgetting in recursive identification through multiple models. In *Proc. Int. J. Control*, pages 1175–1193, 1985.

[2] A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *Int. Journal of Computer Vision*, 1993.

[3] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.

[4] A. Blake, M. Isard, and D. Reynard. Learning to track the visual motion of contours. *J. Artificial Intelligence*, 1995.

[5] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997.

[6] L. Campbell and A. Bobick. Recognition of human body motion using phase space constraints. In *Proc. IEEE Intl. Conf. on Comp. Vision*, 1995.

[7] T. Darrell and A. Pentland. Classifying hand gestures with a view-based distributed representation. In *NIPS*, 1994.

[8] N. Ferrier, S. Rowe, and A. Blake. Real-time traffic monitoring. In *IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, December 1994.

[9] W. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, 1995.

[10] D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[11] M. Isard and A. Blake. A mixed-state CONDENSATION tracker with automatic model-switching. In *Proc. 1998 IEEE International Conf. on Computer Vision*, pages 107–112, Mumbai, India, 1998.

[12] H. Liu. Segmenting and identifying dynamic models from measurement of visual motion. Master's thesis, Department of Mechanical Engineering, University of Wisconsin-Madison, 1999.

[13] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. The MIT Press, 1983.

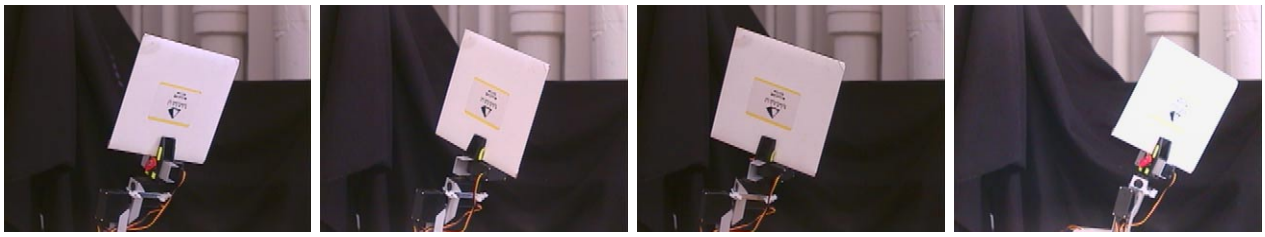
[14] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(6):522–536, 1980.

[15] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proc. IEEE International Conf. on Computer Vision*, Boston, MA, 1995.

[16] S. Seitz and C. Dyer. View-invariant analysis of cyclic motion. *Int. Journal of Computer Vision*, pages 1–23, 1997.

[17] M. Shah and R. Jain. *Model Based Recognition*. Academic Publishers, Boston,MA, 1997.

[18] P. Torr, A. Fitzgibbon, and A. Zisserman. Maintaining multiple motion model hypotheses over many views to recover matching and structure. In *Proc. 1998 IEEE International*



(a) (b) (c) (d)

Figure 4. Four images from the motion sequence used to generate the data in Fig. 6. The target is moved into the field of view, tracking isinitiated, and then the robot commands a cyclic motion. Two oscillating motions are commaned – between views (a) and (b) followed by motion between views (c) and (d).

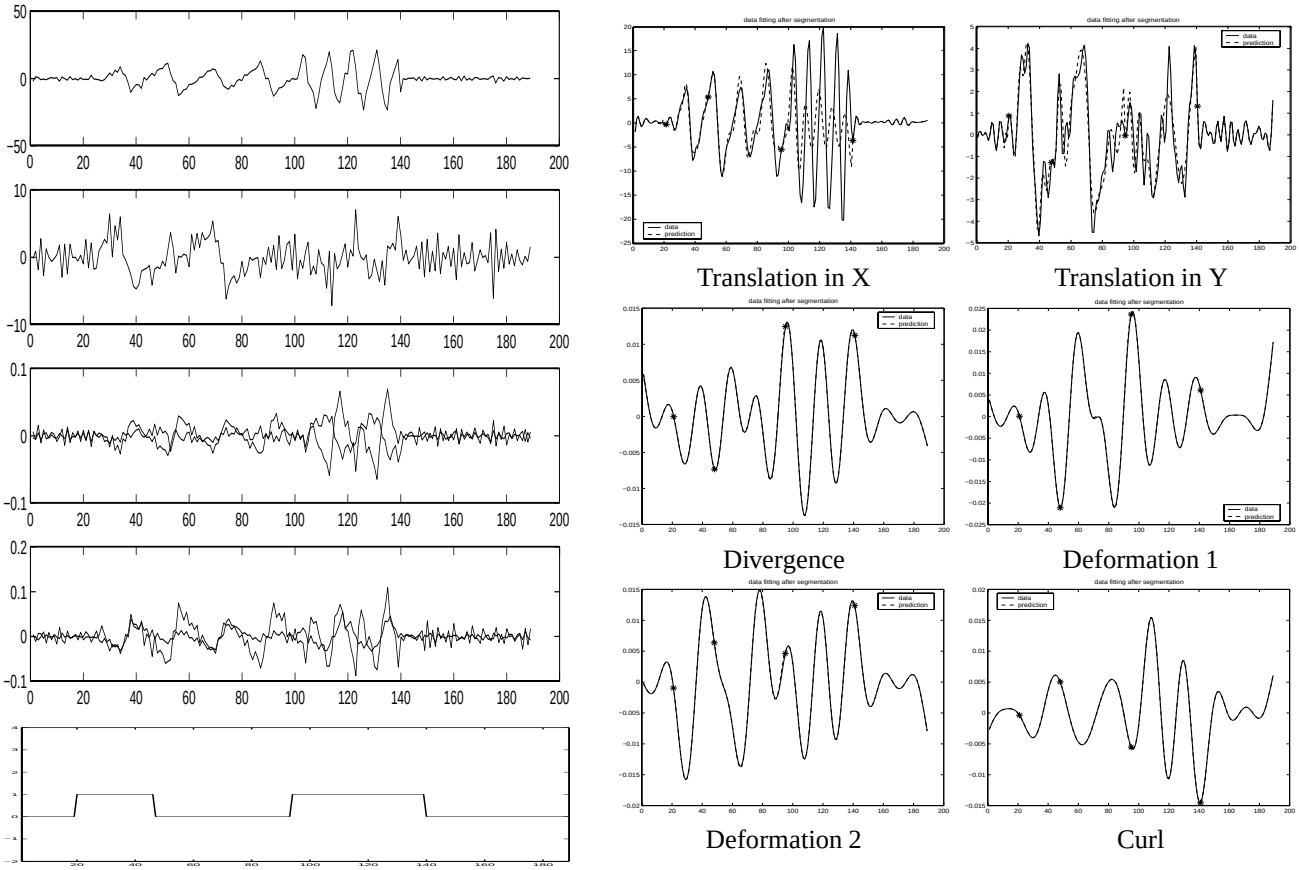


Figure 5. The left hand side plots show measured motion measured (see Fig. 4). From top to bottom: X-translation, Y-translation, divergence and curl, and both components of deformation. The final segmentation is shown beneath the measured motion. The segmentation correctly identifies the two cyclic motions, along with segments for motion prior to, and after, the commanded motions. On the right hand side the results of fitting dynamic models are shown for each of the six affine components as indicated. The measured motion is shown with a solid line, the model prediction with a dashed line, segmentation points astericks.

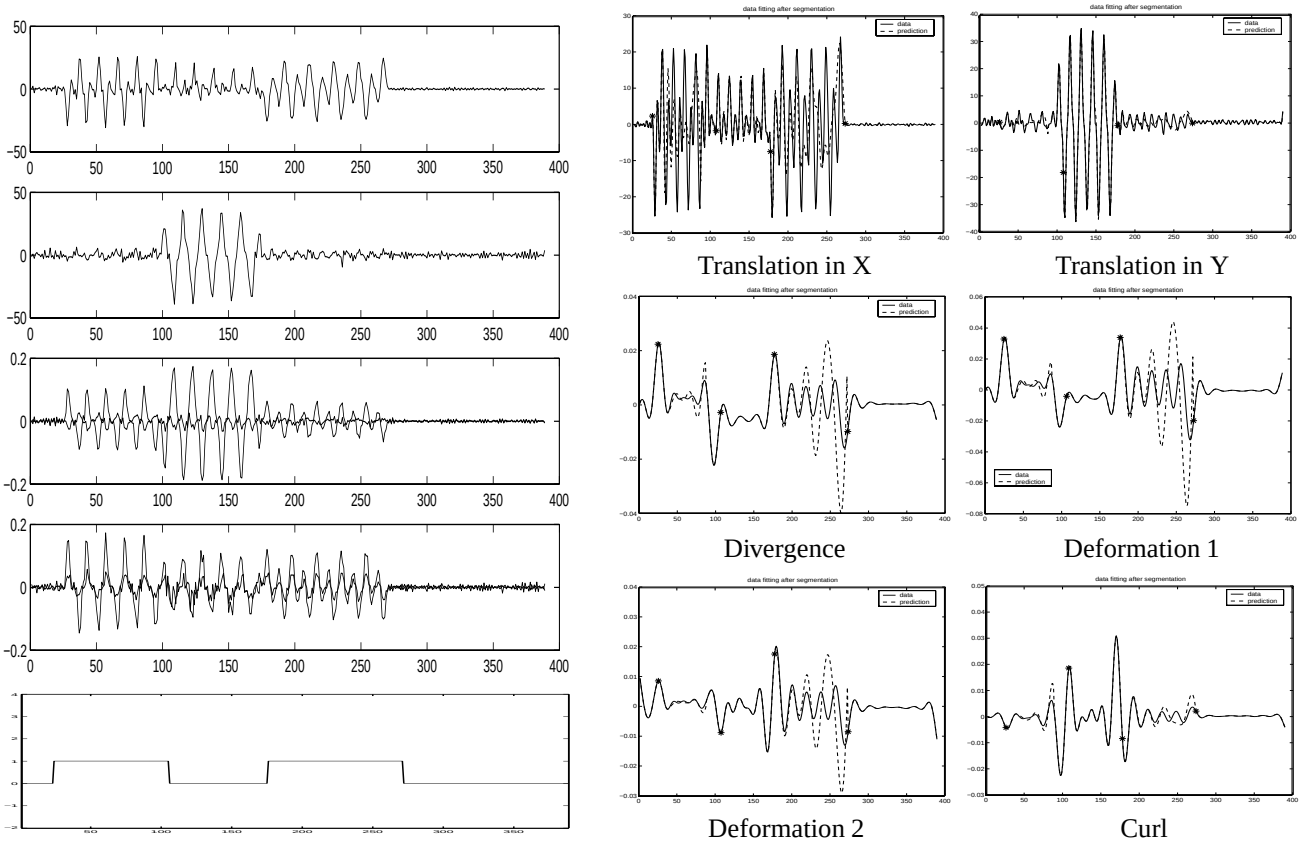


Figure 6. Three cascaded oscillatory motion patterns generated by tracking an object moved by a robot. On the left hand side from top to bottom are plots showing X-translation, Y-translation, divergence and curl, and both components of deformation. The final segmentation is shown beneath the measured motion. On the right hand side the results of fitting dynamic models are shown for each of the six affine components as indicated. The measured motion is shown with a solid line, the model prediction is shown with a dashed line. Error in the fitting can be reduced with further segmentation or by using a higher order AR model.

Conf. on Computer Vision, pages 485–491, Mumbai, India, 1998.

[19] Y. Weiss and E. Adelson. Motion estimation and segmentation using a recurrent mixture of experts architecture. In *IEEE Workshop on Neural Nets for Signal Processing*, Cambridge, MA, 1995.

[20] A. D. Wilson and A. F. Bobick. Recognition and interpretation of parametric gestures. In *Proc. 1998 IEEE International Conf. on Computer Vision*, Mumbai, India, 1998.

[21] Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. In *Proc. 1998 IEEE International Conf. on Computer Vision*, Mumbai, India, 1998.