Motion Mining: Discovering Spatio-Temporal Patterns in Databases of Human Motion

George Kollios
Boston University
gkollios@cs.bu.edu

Stan Sclaroff
Boston University
sclaroff@cs.bu.edu

Margrit Betke
Boston University
betke@cs.bu.edu

Abstract

In the last decade, there has been an explosive growth in the number of computer systems that gather data about human motion via video cameras, magnetic trackers, eye trackers, motion capture body suits and gloves, etc. As these datasets grow, there will be an opportunity to analyze this massive data archive to gain new insights that can be used to improve our understanding and models of human motion. In this paper we discuss new research challenges and possible solutions to data mining problems that appear in such datasets. We believe that a key point on designing efficient and effective solutions for these problems is the interaction of data mining with other disciplines and in particular with computer vision and machine understanding.

1 Introduction

In the last decade, there has been an explosive growth in the number of computer systems that gather data about human motion via video cameras, magnetic trackers, eye trackers, motion capture body suits and gloves, etc. These systems generate streams of 3D motion trajectories or other time series data about human motion that are used in computer human interfaces, computer animation and special effects, analysis of human biomechanics, and surveillance of human activity. Recently, new efforts have formed around the issue of creating archives of human motion data for use as "standard data sets" in the development of new algorithms in the computer science community, as well as for use in studies conducted by researchers from other disciplines (e.g., [13, 17, 18]).

As these datasets grow, there will be an opportunity to analyze this massive data archive to gain new insights that can be used to improve our understanding and models of human motion. Insights gained through *motion mining* could lead to improved methods for computer-assisted physical rehabilitation, occupational safety, and ergonomics, as well as improved methods for sports training, medicine, and diagnosis. Furthermore, motion mining could lead to improved computer vision and pattern recognition algorithms that are specially tuned to basic patterns or clusters found in human motion databases. It could also enable algorithms that automatically recognize anomalous motions because they are outliers when considered as part of the motion database.

Such tools for motion mining and their benefits are still far off. This is because present tools for indexing and searching large databases of human motion data are still in their infancy. Many current methods require searching of descriptive text fields that are entered by hand. Given the spatio-temporal nature of human motion, temporal alignment of the text annotations by a human can be quite laborious and error prone. In addition, text annotations can severely limit what kinds of motion patterns can be found/retrieved in the collected data; this is because such indexing is limited to only those annotations entered. Patterns not annotated or unnoticed are not searchable. To achieve a motion mining system, what is needed are methods to directly index, retrieve, and find patterns in the time series and trajectory stored in databases of human motion.

Database and data mining methods can be used to discover patterns in databases of human motion data. Such data has a spatial-temporal aspect that must be dealt with, and therefore a major issue here is to develop methods for indexing and mining databases of motion trajectories and time series data. Another important problem is to automatically extract and analyze motion data given motion capture or video sequences. Furthermore, another promising direction is to develop tracking/recognition algorithms that can learn from the clusters or patterns of motion found in data mining. Through the use of prior knowledge gained in data mining, it is anticipated that more reliable tracking and recognition algorithms will be achieved.

Spatiotemporal databases can be used to store *representations* or *encodings* of interesting visual events as data records, for example, moving body parts or humans performing an activity. As an alternative, motion capture equipment can be used to directly measure 3D human motion. These encodings will be used for automated data analysis and database indexing and mining.

To analyze video data and create a data record of the visual event in the spatiotemporal database, computer vision techniques are needed. These techniques should identify objects in video sequences and describe and interpret their behaviors in order to support various visual monitoring, communication, and surveillance tasks, such as:

- Classification and motion prediction for objects that appear in a sequence of images,
- Interpretation of a gesture as a communication of a nonverbal computer user with severe disabilities,
- Interpretation of sign language,
- Detection of unusual motion patterns.

The basic architecture of the system is shown in Figure 1. The Computer Vision component takes as input a sequence of images and extracts features for moving objects that appear in the sequence. Examples of such features include the trajectory of each object, the velocity at different time periods, etc. The Data Mining component uses the extracted features to perform data mining operations, such as clustering, outlier detection and indexing. The results of the analysis are provided to a domain expert that should interpret and analyze them further. Also the Computer Vision component uses the feedback from the data analysis component to tune the parameters of its algorithms. Since the size of the datasets is large, we are interested in highly scalable and tunable algorithms. The user should be able to set and change some parameters, depending on the type and nature of the data analysis that s/he wants to perform. On the other hand, the system must provide some default parameters and should be able to work satisfactorily without the user interaction.

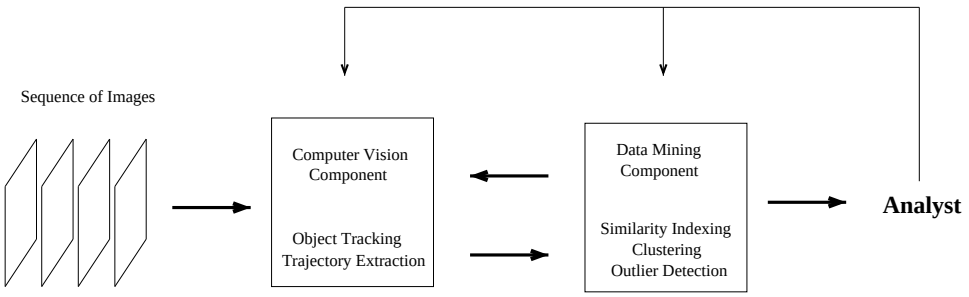

Figure 1: Architecture of the proposed system.

2 Computer Vision

Spatio-temporal indexing relies on robust computer vision algorithms that can detect, estimate, and encode relevant information about human motion in image sequences. We believe that the key to solving these video database retrieval problems is *semantics-preserving image compression*: compact *representations* or *encodings* that preserve essential temporal, spatial, and structural relationships [19, 27]. To be efficient, accurate, and robust, these representations must employ multiple scales in space and time as well as multiple layers of detail. The properties of an object can be represented at many different levels of detail; e.g., by a moving blob of pixels extracted from each image, the object's minimum bounding rectangle (MBR), the object's overall motion trajectory, or detailed parameterizations of the object's shape as it changes over time. Here an object may be a human's full body, and/or body parts including head, hands, feet, eyes, etc.

Computer vision modules are not just needed as a front end of a feed-forward spatio-temporal data management system. The accuracy of the vision algorithms may improve through feedback provided by the *motion mining* modules; the idea is that algorithms allow *retraining* given clusters of data obtained in the Data Mining components of the system, as well as feedback from the analyst (if provided). Furthermore, feedback from the analyst in identifying objects and motions of interest can be used in building classifiers used in identifying events, objects, motions, and changes of interest to the analyst.

2.1 Detecting Moving Blobs and Estimating Their Trajectories

One concern in building our system will be detecting and segmenting changing or moving blobs in image sequences. In the proposed system, we assume that the time-varying image sequences have been registered and rectified to correct for motion of the camera, as well as normalized for differences in imaging conditions. Given a set of registered images, we can make use of change detection and moving blob segmentation methods that rely on first and second order statistics [21, 31] or adaptive mixture models [29].

To estimate the motion trajectory for each blob, we propose to use a predictive tracker [21], which is based on a first order Extended Kalman Filter (EKF) [28, 30]. Each blob's tracker T_i contains information about object location, a binary support map, blob characteristics, MBR, etc. For this application, we can choose an EKF state $\mathbf{x}$ that models the blob's MBR moving along a piece-wise linear trajectory: $\mathbf{x} = (x_0, y_0, x_1, y_1, z\beta, \dot{x}_0, \dot{y}_0, \dot{x}_1, \dot{y}_1, \dot{z\beta})$. In the state vector (x_0, y_0) and (x_1, y_1) are the corners of the MBR, z is the relative distance from the camera, and $\beta = \frac{1}{f}$ is the inverse camera focal length. Note that if the focal length is unknown, this formulation does not provide a unique solution in 3D space. However, the family of allowable solutions all project to a unique solution on the image plane. We can therefore estimate objects' future positions on the image plane and their image trajectories given their motion in $(x, y, z\beta)$ space. Figure 2 illustrates the tracking of walking humans. Occlusion time can be estimated using the EKF predictions and estimates of MBR's velocity and position.

2.2 Estimating Hand or Full Body Pose from Video

Methods for articulated pose estimation have been developed that use machine learning [22, 23]. The approach is general and has been successfully employed in "full-body pose" and "hand pose" estimation problems (see Fig.2).

Given a set of motion capture sequences for training, a set of clusters is built in which each has statistically similar configurations. The forward functions are estimated directly from training data, which in our case are examples of 3D motion capture data of human bodies or human hands.

After training, the system automatically estimates articulated pose parameters from image

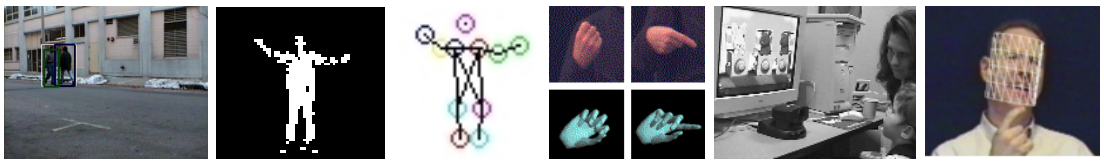

Figure 2: 1. Two people walking and occluding each other. The estimated minimum bounding boxes for each person are shown overlaid on the images. 2. Reconstruction of body pose from silhouettes. 3. Estimated joint locations. 4. Hand pose estimation. 5. Girl with severe disabilites using the Camera Mouse system. 6. Head tracking of a ASL speaker.

features. Given new visual features, an estimate of pose is computed for each cluster via a neural network. From these estimates, the system selects the most likely pose given the learned probability distribution and the visual feature similarity between hypothesis and input. Fig. 2 shows an example of a pose estimate obtained from observing the silhouette of a human subject.

The same approach has been tested in preliminary experiments in tracking human hands. In this case, the mapping functions were trained using approximately 30 3D-hand-motion sequences of American Sign Language (ASL) captured with a Cyberware glove. The parameters captured for the hand model are 22 joint angles. A collection of 200,000 training images was generated using computer graphics by rendering from 86 viewpoints roughly uniformly distributed on the view sphere. Once trained, the system can be used to estimate 3D hand pose as shown in Fig. 2.

2.3 Tracking and Estimating Motion of the Head and Face

Another goal of the proposed system is to develop methods for estimating and encoding head and facial motions. To address this, methods can be developed for automatic and robust detection, tracking, and interpretation of human body components and their motion in video under normal lighting conditions [2, 3, 12]. The system can be used to help people with severe disabilities gain access to a computer and thereby obtain a tool for communication. People with cerebral palsy often cannot speak, have only limited voluntary motions, and so have difficulty communicating with family, friends, and care providers. One early product of this effort is the "Camera Mouse" system [12, 11], which has been developed to provide computer access for people with severe disabilities. The system tracks the computer user's movements with a video camera and translates them into the movements of the mouse pointer on the screen. Fig. 2 shows a thirty-month old user of the Camera Mouse system and her tracked face. Here the vision algorithm is tracking her lower lip.

Another promising method of head tracking employs a texture mapped, 3D computer graphics model [6, 7, 8]. Fig. 2 shows an example of tracking with cylinder head model. The stabilized view of the face obtained via the head tracker has been used as input to an eyebrow raise detector [26, 25].

3 Mining Motion Patterns

Given a spatiotemporal database of human motion, data retrieval and analysis tools are needed. These tools should enable the analyst to effectively access the data in order to get useful information. Therefore, the system must provide appropriate methods to direct the analyst to the right part of the data that may be of his/her interest and minimize the interaction of the analyst with the raw and the intermediate data. To address this problem we use methods and techniques from databases and data mining. The goal of a data mining task is to efficiently find and describe structure and patterns in large datasets. These patterns were previously unknown and not stored explicitly in the database.

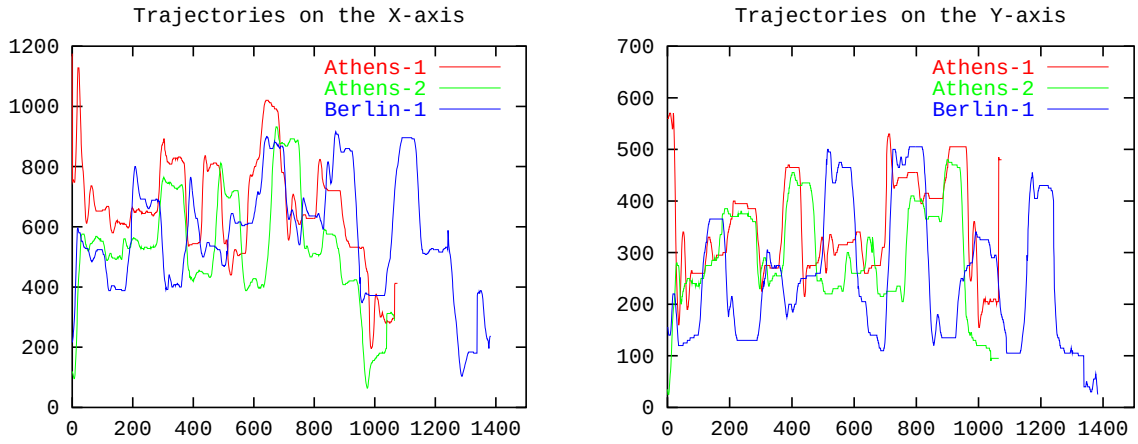

Figure 3: Trajectories of mouse pointer movements that correspond to human facial movements tracked with Camera Mouse system. The motion on X and Y axis during the spelling of two words – Athens (twice) and Berlin (once) are shown.

The first important research issue here is the representation of motion. One approach is to assume that the motion is represented as a sequence of multidimensional points that we call *trajectories*. The main reason for using this representation is its simplicity and generality since every motion pattern can be represented as a time series of points moving in a low dimensional space. Another reason is that simple representations will allow the design of more efficient and robust algorithms. Another approach is to represent the motion using a probabilistic model (for example HMM (Hidden Markov Models)[20] or Semi-Markov Models [10]).

Many data mining tasks require a similarity model (or a distance function) for the objects stored in the database. The problem is not trivial either. Most of the current methods are based on mapping the time-series with n elements to a vector in an n-dimensional space and then use a p-norm distance function to define the similarity measure. The p-norm distance between two n-dimensional vectors $\bar{x}$ and $\bar{y}$ is defined as $L_p(\bar{x}, \bar{y}) = (\sum_{i=1}^{n}(x_i - y_i)^p)^{\frac{1}{p}}$. However this model is inadequate to deal with trajectories. For example, consider trajectories created by humans that move in a similar fashion but with slightly different speeds. The Euclidean distance will fail to capture the similarity. Another problem is that trajectories can have different lengths, which farther complicate the use of the Euclidean distance [9]. A better approach is to use the Longest Common SubSequence ($LCSS$) model [4] or the Time Warping model [1, 24].

In Fig. 3 we show an example of three trajectories of mouse pointer movements that correspond to human facial movements tracked with Camera Mouse system. The Camera Mouse was used with a spelling program. The three trajectories were created when the computer user spelt out some words by moving the mouse pointer to an area on the screen that corresponds to a particular letter and selecting the letter ("clicking" the mouse) by lingering over the area for about a second. The figure represents the output of vision algorithms (data sets) used in evaluating preliminary versions of a data mining system.

In this example, we can see that although two of the sequences are similar we need to allow shift in time and space in order to find a good match. The LCSS model allows shifting in time by its definition. However, we need to allow rigid transformations (translation and rotation) of one sequence in order to find a good match with the other one. The challenge then is how to evaluate such a distance function efficiently.

Next we discuss some research issues on clustering motion data and index these data for answering similarity queries. We assume that the motion is represented using the trajectories of moving objects.

3.1 Indexing Trajectories of Moving Objects

Given some distance measure, we need to design new index methods to store and retrieve trajectories of moving objects using this method. Indexing methods play a very important role in exploratory data mining, where the analyst makes hypotheses about the data and asks queries for validation. Considering the size of the datasets and the on-line nature of the analysis, a system with no indexing capabilities will be of limited use. In a simple scenario, assume that the analyst has identified an interesting pattern. Then s/he can find its motion characteristics and run a query over the database asking for all other objects that moved in the same or similar way. Thus, s/he may be able to find other objects/patterns that are of special interest.

An approach to index a set of trajectories is to embed them into a normed space $\mathcal{D}$ and try to keep the pairwise distances as close as possible to the original ones. Ideally, $\mathcal{D}$ will be a low dimensional Euclidean space $\Re^d$, where d is small.

An interesting embedding method is presented in [16]. The basic idea is to select a set of subsets of S. Let X be a subset of S. Then we find the minimum distance of a given trajectory t to X, $D(t, X) = min_{x \in X} d(x, t)$. This number defines the coordinate of t for the dimension that corresponds to X. Using d number of subsets, $X_1, X_2, \ldots, X_d$, we map each trajectory $t \in S$ to a vector $[D(t, X_1), D(t, X_2), \ldots, D(t, X_d)]$. The distance between two vectors is defined using the l_1 or l_2 norm. For more details about embeddings we refer to [16, 14]. However, the problem with this approach is that the distortion in the pairwise distances can be large and we may have many false negatives.

Another approach is to cluster the set of trajectories and then use the clusters to answer nearest neighbor queries. The quality of the index depends on the distance function used as similarity (or dissimilarity) function. Note that the similarity function that based on LCSS model is not a metric since the triangle inequality does not holds. However we can prove a similar (although weaker) inequality that can be used for pruning some clusters that are far from the query trajectory.

3.2 Motion Clustering and Outlier Detection

A very important data mining task is to cluster set of objects in a large dataset. Therefore, it is important to design clustering methods for large sets of trajectories using various distance functions. The results of a clustering task can be used directly to characterize different groups of objects and summarize their main characteristics. The clusters will be used for re-training the classification and prediction algorithms in the Computer Vision sub-system. Also, hierarchical clustering algorithms can be used for indexing large datasets for similarity queries as we mentioned above.

The only method to cluster trajectory data taking into account the special properties of trajectories has been proposed in [9]. They use a variation of the EM (expectation maximization) algorithm to cluster small sets of trajectories. A problem with this approach is its scalability. Also, the distance measure is based on the probabilistic model which may not be the most appropriate for some specific applications.

Another important task in a data mining system is to identify outliers. An outlier is an object that behaves in an unusual and unpredictable manner. Outlier mining has been used in fraud detection, by detecting unusual usage of credit cards or telecommunication services [5]. In our

type of applications we are interested to find unusual or strange motion patterns. Actually, these patterns are sometimes more interesting for further analysis.

The first issue in outlier detection is to define what data is considered as an outlier for a given dataset. The statistical approach to define outliers is to assume that the distribution of the objects in the dataset follows a specific model and then try to identify objects that deviate from this model. Unfortunately, finding an appropriate model for datasets of trajectories is very difficult and usually real datasets do not follow general statistical models. Another definition of outliers uses a different approach that extends the distance based outlier definition by Knorr and Ng [15]. In particular, given a function that describes a distance between any two objects in the database, we say that an object O is a $DT(k, \xi)$ outlier, if there are at most k objects in the database the have a distance to O smaller than ξ. The challenge then is to find efficiently all the outliers, given some values for k and ξ. An alternative approach is to find the distance of each object to its k-th closest object and report a list of the objects ordered by this distance. Clearly, objects that are "far" from the others will appear first in the list.

References

[1] D. Berndt and J. Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. *In Proceedings of KDD Workshop*, 1994.

[2] M. Betke and J. Kawai. Gaze detection via self-organizing gray-scale units. In *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 70–76, Kerkyra, Greece, September 1999. IEEE.

[3] M. Betke, W. J. Mullally, and J. Magee. Active detection of eye scleras in real time. In *Proceedings of the IEEE Workshop on Human Modeling, Analysis and Synthesis*, Hilton Head Island, SC, June 2000.

[4] B. Bollobas, G. Das, D. Gunopulos, and H. Mannila. Time-Series Similarity Problems and Well-Separated Geometric Sets. *In Proc of the 13th SCG, Nice, France*, 1997.

[5] M. Cahill, F. Chen, D. Lambert, J. Pinheiro, and Don Sun. *Detecting Fraud in the Real World.* Klewer, 2000.

[6] M. La Cascia, J. Isidoro, and S. Sclaroff. Head tracking via robust registration in texture map images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 508–514, June 1998.

[7] M. La Cascia and S. Sclaroff. Fast, reliable head tracking under varying illumination. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 604–610, June 1999.

[8] M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 22(4):322–336, April 2000.

[9] S. Gaffney and P. Smyth. Trajectory Clustering with Mixtures of Regression Models. *In Proc. of the 5th ACM SIGKDD, San Diego, CA*, pages 63–72, August 1999.

[10] X. Ge and P. Smyth. Deformable markov model templates for time-series pattern matching. *In Proc ACM SIGKDD*, 2000.

[11] J. Gips, M. Betke, and P. A. DiMattia. Early experiences using visual tracking for computer access by people with profound physical disabilities. In *Proceedings of the 1st International Conference on Universal Access in Human-Computer Interaction*, New Orleans, LA, August 2001.

[12] J. Gips, M. Betke, and P. Fleming. The Camera Mouse: Preliminary investigation of automated visual tracking for computer access. In *Proceedings of the Rehabilitation Engineering and Assistive Technology Society of North America 2000 Annual Conference*, pages 98–100, Orlando, FL, July 2000.

[13] J. Hodgins. Digital Muybridge: A Repository for Human Motion Data. http://www.interact.nsf.gov/cise/abst.nsf/awards/0079060, May 2000.

[14] G. Hristescu and M. Farach-Colton. Cluster-preserving embedding of proteins, 1999.

[15] E. Knorr and R. Ng. Algorithms for Mining Distance Based Outliers in Large Databases. *In Proceedings of VLDB, New York*, pages 392–403, August 1998.

[16] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. In *In Proc. of the 35th IEEE FOCS*, pages 577–591, 1994.

[17] C. Neidle, S. Sclaroff, and V. Athitsos. Signstream: A tool for linguistic and computer vision research on visual-gestural language data. *Behavior Research Methods, Instruments and Computers*, (submitted in September 2000).

[18] W. Park, X. Zhang, C.B. Woolley, J. Foulke, U. Raschke, and D.B. Chaffin. Integration of magnetic and optical motion tracking devices for capturing human motion data. In *Proc. SAE Human Modeling for Design and Engineering Conference*, 1999.

[19] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision (IJCV)*, 18(3):233–254, June 1996.

[20] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, January:4–16, 1986.

[21] R. Rosales and S. Sclaroff. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. In *Proc. IEEE Workshop on the Interpretation of Visual Motion*, June 1998.

[22] R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume II, pages 721–727, June 2000.

[23] R. Rosales and S. Sclaroff. Learning and synthesizing human body motion and posture. In *Proc. International IEEE Conf. on Automatic Face and Gesture Recognition*, pages 506–511, March 2000.

[24] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-26(1):43–49, February 1978.

[25] S. Sclaroff and J. Isidoro. Active blobs. In *Proc. IEEE international Conf. on Computer Vision (ICCV)*, pages 1146–1153, January 1998.

[26] S. Sclaroff and J. Isidoro. Active blobs: Region-based, deformable appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, (in preparation).

[27] S. Sclaroff and A. Pentland. Modal matching for correspondence and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 17(6):545–561, June 1995.

[28] H.W. Sorenson. Least-Squares Estimation: From Gauss to Kalman. *IEEE Spectrum*, Vol. 7, pp. 63-68, 1970.

[29] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[30] G. Welch and G. Bishop. An Introduction to the Kalman Filter. Technical Report TR 95-041, Computer Science, UNC Chapel Hill, 1995.

[31] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):780-785, 1997.