2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Whole Body Motion Primitive Segmentation from Monocular Video

Dana Kulić, Dongheui Lee and Yoshihiko Nakamura

Abstract— **This paper proposes a novel approach for motion primitive segmentation from continuous full body human motion captured on monocular video. The proposed approach does not require a kinematic model of the person, nor any markers on the body. Instead, optical flow computed directly in the image plane is used to estimate the location of segment points. The approach is based on detecting tracking features in the image based on the Shi and Thomasi algorithm [1]. The optical flow at each feature point is then estimated using the Lucas Kanade Pyramidal Optical Flow estimation algorithm [2]. The feature points are clustered and tracked on-line to find regions of the image with coherent movement. The appearance and disappearance of these coherent clusters indicates the start and end points of motion primitive segments. The algorithm performance is validated on full body motion video sequences, and compared to a joint-angle, motion capture based approach. The results show that the segmentation performance is comparable to the motion capture based approach, while using much simpler hardware and at a lower computational effort.**

I. Introduction

As robots move into human environments, they will need to be able to acquire new skills from non-expert human demonstrators, through observing and imitating human motion. This topic has received considerable attention in the literature [3], [4]. However, in many of the algorithms developed thus far, the number of actions to be learned is specified by the designer, the demonstrated actions are observed and segmented a priori, and the learning is a one shot, off-line process. In this case, there is no need to autonomously segment or cluster the motion data, as this task is performed off-line by the designer. However, to enable robots to learn autonomously over extended periods of time, the robot should be able to observe, segment and classify demonstrated actions on-line during co-location and interaction with the (human) teacher.

In order to extract motion primitives during on-line observation, several key issues must be addressed by the learning system: automated motion segmentation, recognition of previously learned motions, automatic clustering and learning of new motions, and the organization of the learned data into a storage system which allows for easy data retrieval. In our previous work [5], [6], [7] we have been developing algorithms for long-term on-line clustering and organization of whole body motion patterns. Our approach is aimed at incremental learning of motion pattern primitives through long-term observation of human motion. Human motion patterns are abstracted into a stochastic model representation, which can be used for both subsequent motion recognition and generation. As new motion patterns are observed, they are incrementally grouped together based on their relative distance in the model space. In this paper, we focus specifically on the first problem, i.e., the segmentation of the continuous observation sequence into motion segments. In order for the robot to extract meaningful motion primitives from continuous observation of the human demonstrator, the robot must be able to determine the start point and end point of each motion primitive segment. In earlier work [7], we have considered human motion data collected through a motion capture system. The measured locations of the optical markers are used to compute the joint angle data of the human demonstrator via inverse kinematics, and this data is then used for segmentation and motion primitive learning. While motion capture data provides accurate measurements of the human motion, the size and expense of the motion capture system are significant limitations, and prevent the use of such systems for long term data gathering in arbitrary human environments. Therefore, it would be beneficial if the robot could observe and analyze human motion data obtained from on-board sensors, such as a monocular or stereo cameras. In this paper, we develop an algorithm for autonomous segmentation of motion primitives based on monocular camera video alone, which does not make use of joint angle data or a kinematic model of the demonstrator. Instead, the segmentation is based on the analysis of optical flow in the video sequence. The proposed algorithm searches for coherent clusters of optical flow, and generates segmentation points in those frames where there are significant changes to the coherent clusters. Our approach is motivated by models of human vision [8], [9], which postulate that humans have two parallel mechanisms for recognizing biological movements: one based on form and posture analysis (analogous to our previous approach based on joint angle data), and the second based on optical flow motion analysis. Once the data stream is segmented into appropriate motion primitive candidates, the data can be used for motion recognition and generation [10], [11].

A. Related Work

Most motion segmentation algorithms to date consider motion segmentation based on joint angle data. Existing data segmentation algorithms can be divided into two broad categories: algorithms which take advantage of known motion primitives to perform the segmentation, and unsupervised algorithms which require no a-priori knowledge of the motion data to be segmented.

In the first approach, motion primitives are specified by the designer a-priori, and segmentation is based on the

The authors are with the Department of Mechano-Informatics, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8656 Tokyo, Japan {dana,dhlee,nakamura}@ynl.t.u-tokyo.ac.jp

978-1-4244-2789-5/09/$25.00 ©2009 IEEE

comparison between the known motions and the incoming data. Ilg et al. [12] use dynamic programming to find the best match between prototypical motion primitives and an observed sequence based on key features consisting of zero velocity points in key dimensions. In Takano and Nakamura [13], [14] the known motion primitives are encoded via short HMMs. Segmentation points are then decided based on the error between the motion data predicted by the HMM and the actual observed motion data. If the error increases above a certain threshold, a segment point is declared.

In the second type of approach, some assumption is made about the underlying structure of the data at a segmentation point. For example, Matarić and colleagues [15], [16] developed several algorithms for segmenting motions based on the velocity properties of the joint angle vector. In Pomplun and Matarić [15], a segment is recognized when the root mean square (RMS) value of the joint velocities falls below a certain threshold. In this case, the assumption is that there will be a pause in the motion between motion primitives. In Fod et al. [16], it is assumed that there is a change in the direction of movement accompanying a change between motion primitives. Therefore, a segmentation point is recognized when a Zero Velocity Crossing (ZVC) is detected in the joint angle data, in a sufficient number of dimensions. Lieberman and Breazeal [17] improve upon this approach by automating the threshold selection and adding heuristic rules for creating segments when there is contact with the environment. However, with all the velocity based approaches, it becomes more difficult to tune the algorithm as the number of joints increases. For example, it becomes more difficult to select a single threshold for the RMS value of the joint velocities which will accurately differentiate between segments at rest and segments in motions when the dimension space is large and different types of motions (arm motions only vs. motions including arm and leg movement) are considered.

Koenig and Matarić [18] develop a segmentation algorithm based on the variance of the feature data. The algorithm searches for a set of segment points which minimize a cost function of the data variance. In a related approach, Kohlmorgen and Lemm [19] describe a system for automatic on-line segmentation of time series data, based on the assumption that data from the same motion primitive will belong to the same underlying distribution. The incoming data is described as a series of probability density functions, which are formulated as the states of a Hidden Markov Model (HMM), and a minimum cost path is found among the states using an accelerated version of the Viterbi algorithm. However, the algorithm is not tested on a high DoF data stream, such as full body human motion data. Janus and Nakamura [20], [21] and Kulić and Nakamura [7] have applied this approach to human motion capture data, and have shown that the approach is suitable for autonomous motion segmentation with a large number of degrees of freedom.

Human motion tracking from monocular or multiview camera images has been an active area of research in computer vision [22], [23], [24]. Recently, attention has also turned to recognizing human motion and re-targeting it to humanoid robot motion based on monocular or stereo vision. Dariush et al. [25] use a Swiss Ranger sensor to track Cartesian locations of the demonstrator's upper body, and re-target those trajectories to control a humanoid robot via an inverse kinematics controller. Azad et al. [26] use a stereo camera and a kinematic model of the demonstrator to track a demonstrator's motions. In their approach, a particle filter based on shirt color gradient and skin color information is used to estimate the position of the demonstrator relative to the camera and the joint angles of the kinematic model. Lee and Nakamura [11] propose an approach for motion recognition based on optical flow and an existing database of known motions. The system is able to recognize demonstrated motions, and generate motions similar to the demonstrated ones, based on a particle filter simultaneously estimating the demonstrator position relative to the camera frame and finding the best match between the observed optical flow to the optical flow which would be generated by one of the known motions. However, the approaches using monocular or stereo vision proposed so far either assume continuous retargeting [25], or consider only fixed length motion segments [11]. An approach for autonomously segmenting motions directly from the camera video sequence is needed in order to make camera based approaches suitable for motion primitives of variable length, where the length may not be know a priori.

B. Proposed Approach

While the joint angle based approaches can provide accurate segmentation results, they rely on the availability of joint angle data, usually obtained through a motion capture system. Most motion capture systems are bulky and expensive, and are therefore not amenable to use on a mobile platform such as a robot, or in a wide range of environments. Ideally, the robot should be able to extract motion primitives from on-board sensors, such as a monocular or stereo camera. In this paper, we propose an approach for segmenting motion primitives based on optical flow data from a monocular video sequence. First, suitable features to be tracked are identified via the Shi and Tomasi algorithm [1]. The features are then tracked from frame to frame, and their optical flow is estimated, based on the pyramidal version of the Lucas Kanade algorithm [2], [27]. We next perform on-line continuous clustering to identify coherent clusters of optical flow in the image. When a cluster (or a set of clusters) stops or starts moving, or when a cluster exhibits a significant change in direction, a segment point is recorded. Section II describes the details of the proposed algorithm, in Section III the experimental validation of the proposed approach is described, where the algorithm performance is evaluated and compared to joint-based methods. Finally, Section IV provides conclusions and directions for future work.

II. Segmentation based on Optical Flow

The proposed algorithm identifies human motion primitive start and end points based on optical flow from a monocular video sequence. When, a portion of a body is moving, there will be a region in the optical flow image of coherent optical flow, which will appear and disappear with the start and end of the motion. This is similar to the idea of the *kinematic centroid segmentation* algorithm [28], where arms and legs are modeled as pendulums, placing segment boundaries at the beginning and end of pendulum swings. However, unlike Jenkins and Matarić [28], the proposed approach does not rely on a kinematic model, and does not require the explicit identification of the arms and legs in the image.

A. Computation of the Optical Flow

If optical flow is evaluated on the entire image, many of the pixels will contain invalid estimates, due to lack of texture or other unique features around a given pixel, occlusions between the two frames, depth discontinuities or reflection highlights. To reduce computational burden and the number of invalid optical flow values, optical flow is not computed for the entire image. Instead, the image is first searched for good features to track, using the Shi and Tomasi algorithm [1]. This algorithm formally defines the feature's quality based on how well it can be tracked. The tracking problem can be formulated as follows: Given a point $[x_i, y_i]$ (or set of points) in an image I_1, find the point $[x_i + \delta_x, y_i + \delta_y]$ in image I_2 that minimizes the dissimilarity ε:

$$\varepsilon(\delta_x, \delta_y) = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} [I_1(x,y) - I_2(x+\delta_x, y+\delta_y)]^2 \tag{1}$$

where W is the given feature window, and $\delta = [\delta_x \delta_y]^T$ is the displacement vector or the optical flow. To minimize the dissimilarity, equation (1) is differentiated with respect to δ_x and δ_y and set to zero. This equation is then linearized using a 1st order Taylor series expansion to obtain:

$$0 = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (J\delta - (I_1(x,y) - I_2(x,y)) \cdot [\frac{\partial I_2}{\partial x} \frac{\partial I_2}{\partial y}]) \tag{2}$$

where J is the matrix of the image derivaties,

$$J = \begin{bmatrix} (\frac{\partial I}{\partial x})^2 & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & (\frac{\partial I}{\partial y})^2 \end{bmatrix} \tag{3}$$

Equation (2) is the equation of motion used for tracking features from frame to frame. Shi and Tomasi define a good feature to be one which has high eigenvalues of J, which will result in reliable solvability of Equation (2). Features with high eigenvalues imply that the matrix J of the system has a good signal to noise ratio, and that the matrix is well-conditioned. A feature point with two large eigenvalues will usually indicate a corner or a well textured region, i.e., patterns which can be tracked reliably.

Once reliable features are found in the image, the optical flow for those features is computed via the pyramidal version of the Lucas Kanade algorithm [2], [27]. The Lucas Kanade algorithm solves the optical flow equation (Equation 2) iteratively, using Newton-Raphson, however this equation is only valid for small pixel displacements, due the the first order Taylor series approximation. To extend the region of validity, a pyramidal representation of the image is generated, where each level of the pyramid is a down sampled version of the previous level. The optical flow equation is then solved sequentially, starting at the lowest level of the pyramid, and using the lower level solution as the starting value for the iterative solution of the Lucas Kanade optical flow equations. This allows each optical flow residual vector to remain small, even when the overall optical flow magnitude is large.

B. Clustering of the Optical flow data

Once a set of tracking features and the associated optical flow at each feature point is computed, the features are clustered together to find coherent areas of optical flow. Since the data to be clustered is arriving incrementally with each frame, this is an instance of incremental clustering. Many approaches for incremental clustering have been proposed in the literature, including methods based on expectation-maximization [29], Gaussian mixture models [30], [31], vector quantization [32], and hierarchical clustering [33], [34]. We propose a very simple clustering approach here, but other, more sophisticated approaches which can handle a variable number of clusters and explicitly identify when clusters appear or disappear, such as the work of Rodrigues et al. [33], [34], could also be used.

In the proposed approach, the clustering is performed for each frame, using the cluster centroids of the previous frame as the starting points, so that the evolution of coherent areas over time can be tracked. The appearance and disappearance of areas of coherent optical flow indicates are used as indicators of a segmentation point.

First, to eliminate spurious feature points, points with a small magnitude of optical flow are removed from consideration. The remaining points are clustered based on their location in the image plane. Starting with the cluster centers from the previous frame, each point is added to the nearest existing cluster, if the distance between the new point and the nearest cluster center is smaller than D_{max}. Each cluster maintains its current centroid (average position and velocity), which is updated after the addition of a new feature point. If the smallest distance between any existing cluster centroid and the new feature point is larger than D_{max}, a new cluster is instantiated with its centroid at the current feature point.

Cluster maintenance is performed at the start of each frame to remove inactive clusters. A cluster which contains more than K_{min} feature points is considered a valid cluster. A cluster which has had no new points added for N_i frames becomes inactive and is removed. For valid, active clusters, the magnitude and direction of the optical flow are compared to the optical flow in the previous frame. A change of direction occurs when the direction changes by more than α_{min}.

The segmentation estimate is then generated based on changes to the cluster status. A segment point occurs when a valid, active cluster, which has been active more than N_{min} becomes inactive, or when a significant change in the direction of the optical flow occurs. The algorithm is summarized in Figure 1.

```
Initialize cluster set to an empty set
$C \leftarrow \emptyset$
procedure CLUSTERING(Feature Points F)
    for $f \leftarrow F$ do
        Find the shortest dist. between f and the existing clusters
        $d$ = min(dist($f$,$C$))
        $c$ = minindex(dist($f$,$C$))
        if $d < D_{max}$ then
            Add feature f to cluster c
            $c$.add($f$)
        else
            Start a new cluster centered at f
            $c_{new}$.Initialize($f$)
            $C$.add($c_{new}$)
        end if
    end for
end procedure
procedure CLUSTERMAINTENANCE(C)
    for $c \leftarrow C$ do
        Check validity for each cluster
        if $c$.NumFeatures $> K_{min}$ then
            $c$.Valid $\leftarrow$ True
            $c$.FramesActive $\leftarrow$ $c$.FramesActive + 1
        end if
        Check activity for each cluster
        if $c$.NumFeatures = 0 then
            $c$.FramesInactive $\leftarrow$ $c$.FramesInactive + 1
            if $c$.FramesInactive $> N_i$ then
                if $c$.Valid and $c$.FramesActive $> N_{min}$ then
                    $segPoint \leftarrow True$
                end if
                $C$.remove($c$)
            end if
        end if
        Check change of direction for each cluster
        if $c$.Valid and abs($c$.prevDir - $c$.currDir) $> \alpha_{min}$ then
            $segPoint \leftarrow True$
        end if
    end for
end procedure
```

Fig. 1. Optical Flow Clustering and Segmentation Algorithm Pseudocode

C. Parameter Selection

The algorithm contains five parameters: the maximum distance at which a feature is considered for cluster inclusion, D_{max}, the number of features required for cluster validity, K_{min}, the number of frames an empty cluster remains valid, N_i, the minimum change in optical flow direction to consider a change of direction, α_{min}, and the minimum number of frames a cluster must be active before a segment will be considered, N_{min}.

D_{max} determines the largest possible size of a cluster. Making D_{max} larger will tend to reduce the number of segment points overall, since fewer clusters will be formed, as more points can be accommodated in existing clusters. Making D_{max} large will also increase the likelihood that motion from one part of the body switching to another partially overlapping part will not be detected, as the two motions will merge into the same cluster. For example, a segment between a torso movement and an arm movement when the arm partially overlaps the torso will not be detected. However, selecting D_{max} too small will make it likely that a single coherent motion is separated into two or more clusters. D_{max} must therefore be selected relative to the size of the demonstrator on the image plane. In the experiments described below, the maximum intra-cluster distance was selected as 1/4 of the demonstrator height. It is also possible to select this distance programmatically, by considering the standard deviation and the number of points in the existing cluster.

The number of features required for cluster validity, K_{min}, specifies the number of feature points required to form a valid cluster. Setting this parameter greater than 1 is intended to prevent the formation of clusters around spurious feature points, and to ensure that a physical motion is taking place.

N_i, the number of frames an empty cluster remains valid, specifies for how many frames a cluster is maintained in memory even when no new feature points are being added. Setting this variable to a value greater than 1 ensures that a cluster remains active even if features are missed for less than N_i frames. However, increasing this variable also introduces a delay in segment point recognition. This variable becomes important if there are temporary occlusions or other disturbances in the image that would make the optical flow temporarily unavailable. In the sequences tested below, this was not an issue, so the parameter was not used.

The parameter α_{min} specifies the magnitude of direction change required for a segment point to be inserted. However, care must be taken not to select too small of an angle, as there is significant variability in the average optical flow direction from frame to frame, due to errors in estimation of individual feature optical flow. In particular, this parameter is sensitive to the orientation of the moving body part relative to the camera. Changes in direction which are orthogonal to the camera viewing axis can easily be detected via this parameter, while changes in direction which are parallel to the camera viewing axis may not be detected.

The parameter N_{min} specifies the minimum number of frames a cluster must be active before its disappearance is considered a segment point. The purpose of this parameter is to eliminate false positives due to spurious segment points when a cluster is visible for only a short amount of time, for example due to changes in occlusion. This parameter determines the minimum length of a valid motion sequence. The type and speed of motions performed must also be considered when selecting this parameter, especially if fast, short motions are expected.

In the current implementation, the parameters are set to constant values, however, they could also be optimized automatically in a supervised training phase, or, in the

TABLE I

ALGORITHM PARAMETERS

Parameter	Value
D_{max}	70 pixels
K_{min}	3
N_i	1
α_{min}	90 deg
N_{min}	15

case of D_{max}, computed autonomously. Table I summarizes the parameters used for the algorithm in the experiments described below.

III. EXPERIMENTS

The algorithm described in Section II was implemented based on the Intel OpenCV library [35]. The algorithm was tested on a video sequence capturing a demonstrator performing a lengthy continuous sequence of a variety of whole body motions. The camera captured data at 30 frames per second. The demonstrator's motions were captured via an optical motion capture system at the same time [7], to allow a direct comparison between the performance of the proposed algorithm and motion capture based approaches. The motion capture data was also animated in slow motion and segmented manually, for comparison with the automated segmentation results.

The data set consists of 17 minutes of continuous whole body motion data of a single human subject. During the data sequence, the subject performs a variety of full body motions, including a walk in place motion, a squat motion, kicking and arm raising. The subject performs a total of 751 motion segments, as determined by the manual segmentation. In some cases, there is a pause between motions, while other motions are fluidly connected.

Figure 2 shows a visualization of the segmentation algorithm performance during a sample motion. Prior to the start of the motion, while the subject is still, no feature points exhibit significant optical flow. As the demonstrator's arm begins to move, feature points with significant optical flow are detected; the feature point locations and associated optical flow are shown with red arrows in Figure 2. The feature points are clustered and the cluster centroid estimated for each frame, shown as a green circle and arrow in Figure 2. The size of the cluster is indicated as one standard deviation. When a valid, active cluster becomes inactive, or changes direction, a segmentation point is indicated with a blue circle. The blue circle is located at the last stored position of the inactivated cluster. As can be seen from the figure, although individual feature point optical flow estimates can contain significant error, the aggregate cluster measure is quite stable, and a reliable indicator of segment point location.

The image processing was implemented on a Intel Xeon 5150 2.66GHz processor. The maximum measured computation time per frame is 17ms, which is well below the frame rate of 30fps, indicating that the algorithm can be used to perform the segmentation online.

TABLE II

SEGMENTATION RESULTS FOR THE BASIC AND MODIFIED ALGORITHMS

Algorithm	Correct	False Pos	False Neg
Joint Angle Based	594	240	158
Optical Flow Based	596	139	155

Table II compares the performance of the proposed segmentation algorithm to a stochastic joint angle based approach [7], using the manually segmented results as ground truth. The stochastic approach makes use of a kinematic model to convert marker data obtained from motion capture to joint angle data. A segment point was considered correct if it occurred within 5 frames of the manually obtained results. A segment point was counted as a false positive, if it occurred in a section where no manual segment point was specified within 5 frames of the given segment point. A false negative was counted if no segmentation point was specified within a 5 frame window of a manually found segmentation point. As can be seen in Table II, the algorithm achieves a correct segmentation rate comparable the joint angle based approach. The basic correct segmentation result is equivalent to the joint angle based approach, while producing significantly fewer false positives. In addition, the optical flow approach has much simpler hardware and computational requirements, and does not require a kinematic model of the demonstrator.

Similar to the joint angle segmentation result [7], some false positive errors are a result of the segment definition used in the manual segmentation, when distinctive subsegments occur within a segment which are not marked in the manual results. For example, on some occasions, within the right punch extend (RPE) segment, two clear subsegments can be observed: raising both hands into a fighting stance, and then extending the right hand. At other times, the RPE segment is executed as a single smooth movement, with no discernible subsegments.

Figure 3 shows the segmentation results classified by motion type. As can be seen from Figure 3, the best segmentation results are achieved when a single body part is moving, or there is a synchronous motion of two body parts, such as with a single arm raise motion, or a both arms raise motion. The worst results are obtained for segmenting the end of a squat(SQU) or bow (BAU) movement. Both of these movements result in a cluster of movement centered on the torso. Following the end of a torso movement, if the subsequent movement is an arm movement, it is likely that the arm will be classified as belonging to the same cluster, thus resulting in a continuation of the same cluster, and a false negative error. This suggests that these errors could be corrected by additional analysis of the cluster shape over time, for example by introducing a segmentation point if there is a significant change to the size of the cluster.

A second type of error occurs with the kick motion (LKE to LKR and RKE to RKR) segmentation. A kick motion is a very fast, smooth motion, such that there are no frames with no optical flow when the motion switches from kick extend to kick retract. The segment point should be easily

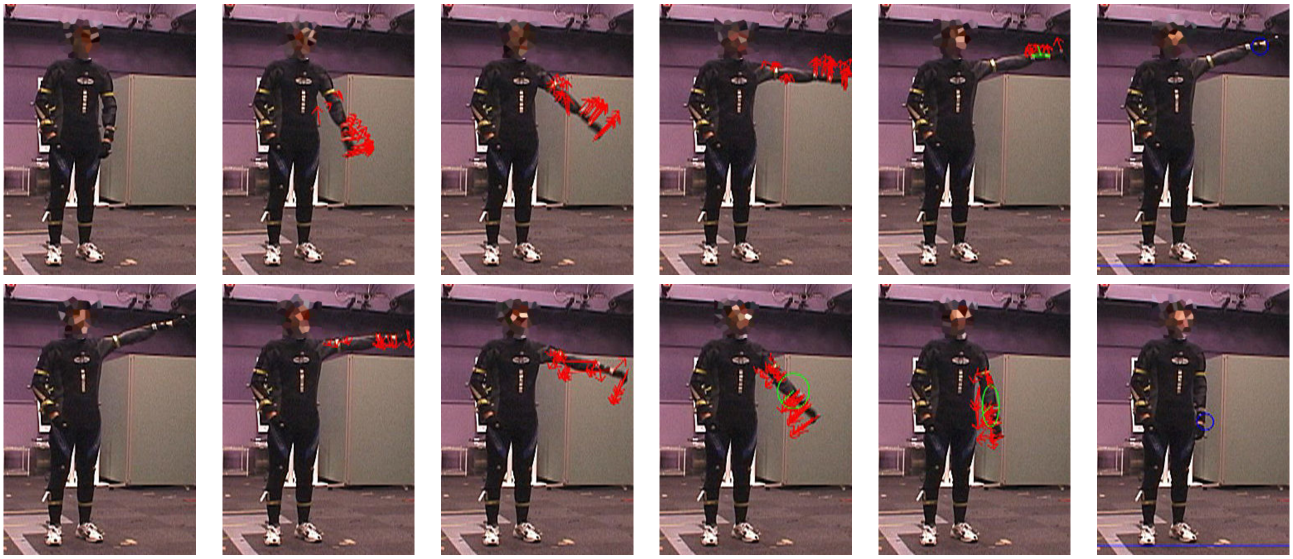

Fig. 2. Exemplar segmentation results for the left hand raise motion. Red arrows indicate feature points with significant optical flow; cluster centroids of all active clusters are shown as a green circle and arrow; a blue circle indicates when a valid, active cluster becomes inactive, i.e. a segmentation point (visible by the left hand in the final frame of each row).

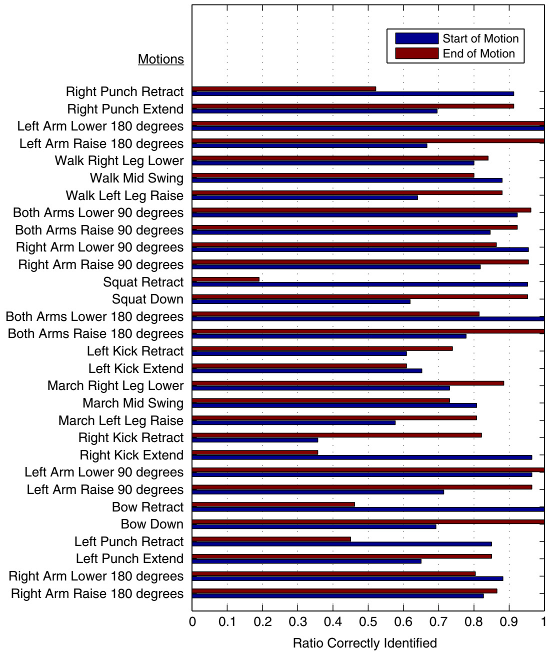

Fig. 3. Segmentation results by motion type

detected based on the change in direction of the two motions. However, in this video sequence, the demonstrator faces the camera, such that the change in direction of the kick motion is parallel to the camera axis, and thus not easily discernible based on the change of direction measure.

For motions where the bulk of the motion occurs orthogonal to the camera axis, the proposed algorithm achieves stable, reliable segmentation results with few false positives. The proposed optical flow segmentation algorithm achieves a similar error rate, and a better overall false positive error rate than the joint angle based algorithm, even though significantly less information is available.

IV. Conclusions and Future Work

In order to learn and extract behaviors and motion primitives from continuous, on-line observation, robots must be able to autonomously extract and segment motion primitive candidates. This paper proposes a novel approach for motion segmentation, based on monocular video sequence data. Rather than using a kinematic model of the actor and extracting joint angle data for use in segmentation, the algorithm operates on the image data directly, and extracts segment points based on changes in the optical flow of the image. Feature points which are suitable for tracking and which exhibit significant optical flow are extracted from the image, and clustered into coherent groups, in terms of position and optical flow direction and magnitude. The cluster set is updated from frame to frame. Changes to the cluster set, such as cluster disappearance or change of optical flow direction are used as indications that a motion primitive has started or ended. The proposed algorithm was tested on an extended video frame sequence, and compared to a joint angle based method. The proposed algorithm achieves comparable performance to previous methods, while requiring a much simpler hardware setup, and using significantly less computational resources.

In future work, further improvements to the segmentation algorithm will be considered, such as improving segmentation by tracking the cluster behavior such as the cluster size or direction over a longer time frame, rather than simply the difference between two adjacent frames. The system will also be tested in more complex environments, and including changes in orientation and location of the demonstrator. The segmentation algorithm will also be integrated with motion recognition and generation [10], [11] to enable continuous, fully autonomous on-line learning and motion recognition during co-location and interaction with a human demonstrator.

V. ACKNOWLEDGMENTS

This work is supported by the Japanese Society for the Promotion of Science Category S Grant-in-Aid for Scientific Research 20220001.

REFERENCES

[1] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[2] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Image Understanding Workshop*, 1981, pp. 121–130.

[3] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Sciences*, vol. 6, no. 11, pp. 481–487, 2002.

[4] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 358, pp. 537 – 547, 2003.

[5] D. Kulić, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains," *International Journal of Robotics Research*, vol. 27, no. 7, pp. 761–784, 2008.

[6] ——, "Combining automated on-line segmentation and incremental clustering for whole body motions," in *IEEE International Conference on Robotics and Automation*, 2008, 2591–2598.

[7] D. Kulić and Y. Nakamura, "Scaffolding on-line segmentation of full body human motion patterns," in *IEEE/RJS International Conference on Intelligent Robots and Systems*, 2008, pp. 2860–2866.

[8] G. Johansson, "Visual motion perception," *Scientific American*, pp. 76–88, 1975.

[9] M. A. Giese and T. Poggio, "Neural mechanisms for the recognition of biological movements," *Nature Reviews: Neuroscience*, vol. 4, pp. 179–192, 2003.

[10] D. Lee and Y. Nakamura, "Mimesis scheme using a monocular vision system on a humanoid robot," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 2162–2167.

[11] ——, "Motion capturing from monocular vision by statistical inference based on motion database: Vector field approach," in *IEEE International Conference on Intelligent Robots and Systems*, 2007, pp. 617–623.

[12] W. Ilg, G. H. Bakir, J. Mezger, and M. A. Giese, "On the representation, learning and transfer of spatio-temporal movement characteristics," *International Journal of Humanoid Robotics*, vol. 1, no. 4, pp. 613–636, 2004.

[13] W. Takano and Y. Nakamura, "Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 425–431.

[14] W. Takano, "Stochastic segmentation, proto-symbol coding and clustering of motion patterns and their application to signifiant communication between man and humanoid robot," Ph.D. dissertation, University of Tokyo, 2006.

[15] M. Pomplun and M. J. Matarić, "Evaluation metrics and results of human arm movement imitation," in *IEEE-RAS International Conference on Humanoid Robotics*, 2000.

[16] A. Fod, M. J. Matarić, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, no. 1, pp. 39–54, 2002.

[17] J. Lieberman and C. Breazeal, "Improvements on action parsing and action interpolatin for learning through demonstration," in *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, 2004, pp. 342–365.

[18] N. Koenig and M. J. Matarić, "Behavior-based segmentation of demonstrated tasks," in *Proceedings of the International Conference on Development and Learning*, 2006.

[19] J. Kohlmorgen and S. Lemm, "A dynamic hmm for on-line segmentation of sequential data," in *NIPS 2001: Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14, 2002, pp. 793–800.

[20] B. Janus and Y. Nakamura, "Unsupervised probabilistic segmentation of motion data for mimesis modeling," in *IEEE International Conference on Advanced Robotics*, 2005, pp. 411–417.

[21] B. Janus, "On-line motion segmentation algorithm for mimesis model," Master's thesis, University of Tokyo, 2006.

[22] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3d human figures using 2d image motion," in *European Conference on Computer Vision*, 2000, pp. 702–718.

[23] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," in *IEEE Conference on Vision and Pattern Recognition*, 2004, pp. 421–428.

[24] R. Kehl and L. V. Gool, "Markerless tracking of complex human motions from multiple views," *Computer Vision and Image Understanding*, vol. 104, pp. 190–209, 2006.

[25] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura, "Whole body humanoid control from human motion descriptors," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 2677–2684.

[26] P. Azad, A. Ude, T. Asfour, and R. Dillmann, "Stereo-based markerless human motion capture for humanoid robot systems," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 3951–3956.

[27] J. J. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm," Intel Corporation, Tech. Rep., 2002.

[28] O. C. Jenkins and M. Matarić, "Performance-derived behavior vocabularies: Data-driven acquisition of skills from motion," *International Journal of Humanoid Robotics*, vol. 1, no. 2, pp. 237–288, 2004.

[29] P. S. Bradley, U. M. Fayyad, and C. A. Reina, "Scaling em (expectation-maximization) clustering to large databases," Microsoft Research, Tech. Rep., 1999.

[30] O. Arandjelovic and R. Cipolla, "Incremental learning of temporally-coherent gaussian mixture models," in *British Machine Vision Conference*, 2005, pp. 759–768.

[31] M. Song and H. Wang, "Highly efficient incremental estimation of gaussian mixture models for online data stream clustering," in *Proceeding of the Society of Photonics, Optics and Imaging*, 2005, pp. 174–183.

[32] E. Lughofer, "Extensions of vector quantization for incremental clustering," *Pattern Recognition*, vol. 41, pp. 995–1011, 2008.

[33] P. P. Rodrigues, J. Gama, and J. P. Pedroso, "Hierarchical clustering of time series data streams," *IEEE Transactions on Knowledge and Data Engineering*, 2008, to Appear.

[34] ——, "Odac: Hierarchical clustering of time series data streams," in *SIAM International Conference on Data Mining*.

[35] *Open Source Computer Vision Library*, Intel Corporation.