Ego-motion Classification for Body-worn Videos

Zhaoyi Meng, Javier Sánchez, Jean-Michel Morel, Andrea L Bertozzi, P. Jeffrey Brantingham

Abstract Portable cameras record dynamic first-person video footage and these videos contain information on the motion of the individual to whom the camera is mounted, defined as ego. We address the task of discovering ego-motion from the video itself, without other external calibration information. We investigate the use of similarity transformations between successive video frames to extract signals reflecting ego-motions and their frequencies. We use novel graph-based unsupervised and semi-supervised learning algorithms to segment the video frames into different ego-motion categories. Our results show very accurate results on both choreographed test videos and ego-motion videos provided by the Los Angeles Police Department.

1 Introduction

Affordable high-quality cameras for recording the first-person point-of-view experience, such as GoPro, are an increasingly common item in many aspects of people's lives. In this paper, we present a novel approach for segmenting or indexing body-worn videos to different ego-motion categories.

Prior work on vision-based first-person human action analysis has focused a lot on indoor activities, such as object recognition [23], hand gesture recognition [17] [31], sign language recognition [28], context aware gesture recognition [27], hand tracking [30] and detecting daily life activities [22]. Work with body-worn sensors has also been shown to be effective for categorizing human actions and activities [10] [26]. An unsupervised ego-action learning method was proposed in [13] for sports videos. The basis of video indexing is to model the transformation between

Zhaoyi Meng
University of California, Los Angeles, e-mail: mzhy@ucla.edu

Javier Sánchez
Universidad de Las Palmas de Gran Canaria e-mail: jsanchez@ulpgc.es

successive frames in the video. For the purpose of video indexing, several studies have examined parametric models of frame transformation such as [6],[12]. Parametric models can be also used for video stabilization [25], and panorama construction [11].

In this paper, we propose an approach to classify different ego-motion categories. We know that human motion observed from a first-person point-of-view can be captured by the global displacement between successive frames. This means that we should be able to aggregate global motion and marginalize out local outlier motion. We also know that motion involving the human gait has an inherent frequency component. Therefore we can expect that frequency analysis can be used as an important feature for ego-action categorization. We propose the use of a parametric model for calculating the simple global representation of motion. This approach produces a low dimensional representation of the motion of the ego. We then classify the ego-motion using novel graph-based semi-supervised and unsupervised learning algorithms. The algorithms are motivated by PDE-based image segmentation method and achieve high performance in both accuracy and efficiency for different discrete data sets.

We consider the ego-motion classification problem with both benchmark and real-world data. Working with both types of data is critical because of the stark differences in the degree of difficulty in the analysis of video data collected under controlled and uncontrolled or "wild" conditions. Benchmark datasets with known ground truth are developed under experimental conditions controlled by the researcher. Such datasets attempt to simulate the types of behaviors that are of most interest to the researcher. Simulations may favor positive outcomes because they seek not only to limit sources of error linked to video image quality, but also enhance target behaviors of interest. For example, experimental protocols that seek to enhance camera stability, ensure adequate lighting conditions, avoid obstructions may all assist in the algorithmic task. Ensuring that experimental participants enact well-defined or discrete transitions between different types of behavior, or exaggerate the differences between behavioral modes may favor accurate segmentation. We draw on choreographed video collected under controlled circumstances to develop our approach.

Videos not collected under controlled conditions may nevertheless be hand-labeled by the researcher to produce a ground truth. Such videos may be subject to many more quality challenges than simulated scenes. Actual behavior and conditions as they exist on the ground are unforgiving. People in real-world settings may not act in discrete, linear sequences, nor are they necessarily inclined to exaggerate their different actions for easy detection. Ego-motions may also proceed so quickly that they defy discrete recognition. We may also lack sufficient semantic categories to capture the diversity of real-world behavior. Real-world video systems may also not be state-of-the-art and therefore suffer from poor camera stability, low frame rate, low resolution, poor color saturation and data collection errors (both human and mechanical). All of these effects can drastically impact the ability of the researcher to label video for ground truth, which introduces errors into algorithmic methods. We draw on police body-worn video (BWV) to evaluate how our methods

perform under challenging real-world conditions. Police BWV is typically shaky, contains noise from low light conditions, poor color saturation and occlusions, and represents diverse and often mixed motion routines.

The paper is organized as follows. In Section 2, we describe the method for motion feature extraction for two successive frames. In Section 3, we investigate the semi-supervised and unsupervised graph-based MBO algorithms for classification. In Section 4, we elaborate on our experimental results using choreographed test video and real-world video data. Section 5 concludes the paper.

2 Motion Features

We characterize motion in a video sequence using a set of features. The features represent the relative movements of ego, the individual on whom the video camera is mounted. The features depend on the estimation of parametric models between successive frames and on the analysis of periodic signals of the motion through characteristic frequencies. We illustrate our method of constructing the motion features in Figure 1. In subsection 2.1, we discuss how to use the inverse compositional algorithm to estimate the similarity transformation between successive frames. This transformation is represented by four parameters t_x, t_y, a and b. In subsection 2.2, we construct four of the features to be used for the video segmentation – horizontal displacement (x), vertical displacement (y), angle of rotation (r) and zoom (z) using the similarity transformation. In addition, the characteristic frequencies of these four signals are computed using the method discussed in subsection 2.3. In subsection 2.4, we combine the four movement features and four frequency features to obtain the eight-dimensional feature vector for each transformation between two successive frames. It is this feature vector that will be used for the graph-based machine learning method.

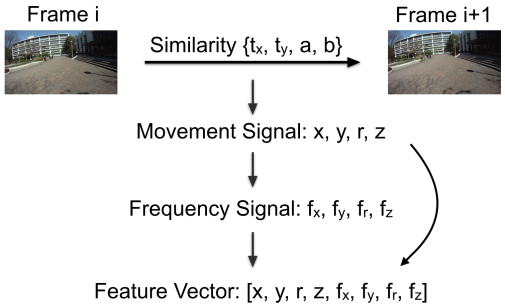

Fig. 1 The process of constructing the motion features for each two successive frames.

2.1 Transformations between two Successive Frames

To compute the motion of the video sequence, we estimate the similarity transformations between consecutive frames using the inverse compositional algorithm [2, 1]. It is possible to use more general parametric motions, such as affinities or homographies. However, the calculation of these is more prone to errors when some camera shake is present. In any case, we find that the four parameters of the similarity are sufficient to characterize motion.

The inverse compositional algorithm is an improvement of the Lucas-Kanade method [14, 1] for image registration. Its implementation in [24] includes the use of robust error functions, which allows estimating the correct transformation even in the presence of occlusions or multiple motions. Let $\mathbf{I}_1(\mathbf{x})$ and $\mathbf{I}_2(\mathbf{x})$ be two images, with $\mathbf{x} = (x, y)$. Let $\mathbf{p}$ be the global displacement vector between the two images and $\Delta\mathbf{p}$ be the incremental displacement vector at each iteration. Let $\mathbf{x}'(\mathbf{x};\mathbf{p},\Delta\mathbf{p})$ be the correspondence map from the left to the right image, or equivalently two frames in a video sequence, parameterized by $\mathbf{p}$ and the incremental refinement $\Delta\mathbf{p}$. The energy model is given by

$$E(\Delta\mathbf{p}) = \sum_{\mathbf{x}} \rho\left(\left|\mathbf{I}_2(\mathbf{x}'(\mathbf{x};\mathbf{p})) - \mathbf{I}_1(\mathbf{x}'(\mathbf{x};\Delta\mathbf{p}))\right|_2^2;\lambda\right), \tag{1}$$

where $\rho(\cdot)$ is a function that gives less weight to large values of the argument, where the difference in image intensities is big (e.g., $\rho(s^2,\lambda) = 0.5s^2/(s^2+\lambda^2)$).

Minimizing the energy with respect to $\Delta\mathbf{p}$ yields:

$$\Delta\mathbf{p} = H_\delta^{-1} \sum_{\mathbf{x}} \rho' \cdot (\nabla\mathbf{I}_1(\mathbf{x})\mathbf{J}(\mathbf{x}))^T \left(\mathbf{I}_2(\mathbf{x}'(\mathbf{x};\mathbf{p})) - \mathbf{I}_1(\mathbf{x})\right), \tag{2}$$

with

$$\begin{aligned} H_\delta &= \sum_{\mathbf{x}} \rho' \cdot (\nabla\mathbf{I}_1(\mathbf{x})\mathbf{J}(\mathbf{x}))^T \nabla\mathbf{I}_1(\mathbf{x})\mathbf{J}(\mathbf{x}) \\ &= \begin{pmatrix} \sum_{\mathbf{x}} \rho' \cdot (\mathbf{I}_{1,x}(\mathbf{x})\mathbf{J}(\mathbf{x}))^T \mathbf{I}_{1,x}(\mathbf{x})\mathbf{J}(\mathbf{x}) & \sum_{\mathbf{x}} \rho' \cdot (\mathbf{I}_{1,x}(\mathbf{x})\mathbf{J}(\mathbf{x}))^T \mathbf{I}_{1,y}(\mathbf{x})\mathbf{J}(\mathbf{x}) \\ \sum_{\mathbf{x}} \rho' \cdot (\mathbf{I}_{1,x}(\mathbf{x})\mathbf{J}(\mathbf{x}))^T \mathbf{I}_{1,y}(\mathbf{x})\mathbf{J}(\mathbf{x}) & \sum_{\mathbf{x}} \rho' \cdot (\mathbf{I}_{1,y}(\mathbf{x})\mathbf{J}(\mathbf{x}))^T \mathbf{I}_{1,y}(\mathbf{x})\mathbf{J}(\mathbf{x}) \end{pmatrix}, \end{aligned} \tag{3}$$

and $\rho' := \rho'\left(|\mathbf{I}_2(\mathbf{x}'(\mathbf{x};\mathbf{p})) - \mathbf{I}_1(\mathbf{x})|_2^2;\lambda\right)$. $\mathbf{J}(\mathbf{x};\mathbf{p}) = \frac{\partial\mathbf{x}'(\mathbf{x};\mathbf{p})}{\partial\mathbf{p}}$ is the Jacobian of the transformation. Table 1 lists the similarity transformation and its Jacobian using the parametrization proposed in [32].

The minimum of this energy provides the parameters of the transformation. To reach a highly accurate solution, the algorithm uses an iterative process. It also includes a coarse-to-fine strategy for estimating large displacements. See [24] for further details.

Transform	**Parameters – p**	**Matrix – H(p)**	**Jacobian – J(x;p)**
Similarity	(t_x, t_y, a, b)	$\begin{pmatrix} 1+a & -b & t_x \\ b & 1+a & t_y \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & x & -y \\ 0 & 1 & y & x \end{pmatrix}$

Table 1 Similarity transformation and its Jacobian

2.2 Movement Signal

Simple motions, such as horizontal (x) and vertical (y) movements, zoom (z) and rotation (r) information can be computed given the similarity. The procedure for calculating the displacement of the central pixel is shown in Algorithm 1.

Algorithm 1: Calculate the displacement of the central pixel

Input : The similarity $\mathbf{H}$, size of the frame n_x and n_y
Output: x,y
1: $\mathbf{p_m} \leftarrow (n_x/2, n_y/2, 1)^T$ {the center of the frame}
2: $(p_1, p_2, p_3)^T \leftarrow \mathbf{H} \cdot \mathbf{p_m}$ {project the center point using the similarity }
3: $(p_1, p_2, p_3)^T \leftarrow (p_1, p_2, p_3)^T / p_3$ {normalize by the third component}
4: $x \leftarrow p_1 - n_x/2$ {the horizontal movement}
5: $y \leftarrow p_2 - n_y/2$ {the vertical movement}
6: **return** x, y

Since the similarity includes the composition of a zoom and rotation matrices, it is easy to obtain these coefficients from the parametrization of Table 1. In this case, the rotation and zoom factor are calculated as

$$r = \arctan\left(\frac{b}{1+a}\right),\ z = \sqrt{(1+a)^2 + b^2}, \tag{4}$$

respectively.

The signals from raw video footage may have abnormally large values. We filter out these values in preprocessing. We replace the signal value by μ, where μ is the mean of the signal sequence and σ is the standard derivation, if the signal value is outside the $(\mu - 3\sigma, \mu + 3\sigma)$ region. The filtered signals can still be very noisy. We use convolutions with a Gaussian function to smooth these signals, which is the basic idea in video stabilization [25].

We use the QUAD video data set [1] to examine ego-motion signals. We discuss the details of this data set in section 4.

[1] The data set can be found at: http://www.cs.cmu.edu/~kkitani/datasets/

The motion signals we calculate using Algorithm 1 and Equation (4) are shown in Figure 2. The left column gives the raw data x, y, z and r and the right column the corresponding filtered and smoothed data.

The periodic pattern correlates with the periodic actions in the QUAD video. The large oscillation of x corresponds to ego turning left and right repeatedly. The large oscillation of y corresponds to ego repeatedly looking up and looking down. The four peaks in z correspond to ego walking and running, since the frames zoom fast when the person is walking or running. The large oscillations of rotation r also correlate with the movements of turning left, turning right, looking up and looking down.

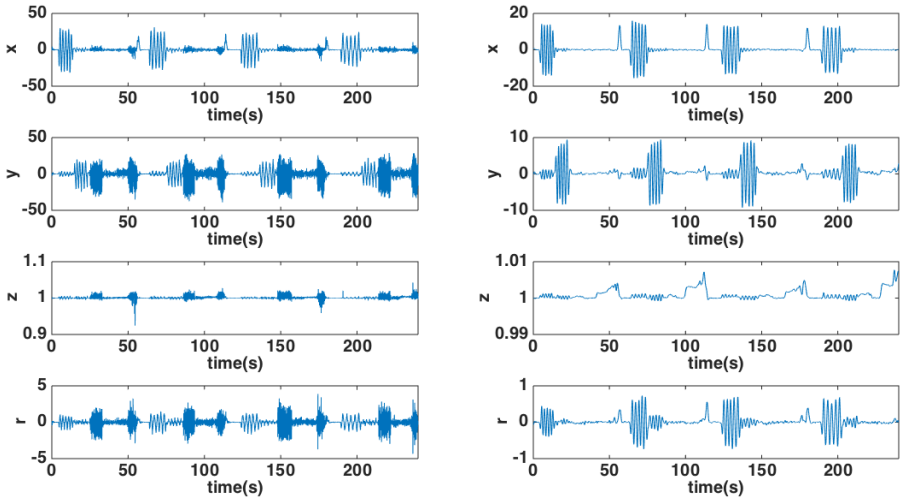

Fig. 2 The x, y, r and z signals. On the left, the original signals and, on the right, the corresponding filtered and smoothed data.

2.3 Frequency Signal

Some ego-motions are periodic, such as jumping, walking and running. Periodic motions have different characteristic frequencies. This observation leads us to investigate the frequencies of x, y, z and r using Fourier analysis. We use the short-time Fourier transform (STFT) to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. In practice, the procedure for computing STFTs is to use a sliding window of fixed length and compute the Fourier transform as the window slides over the whole signal. We use the Hann window here:

$$w(n) = 0.5\left(1 - cos(\frac{2\pi n}{N-1})\right). \tag{5}$$

As shown in Figure 3, the Hann window is zero at the boundaries which reduces the artifacts at the boundary. The STFT is defined by:

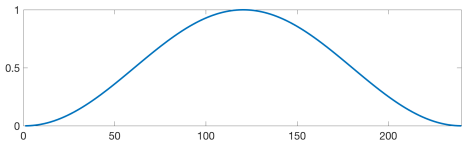

Fig. 3 The Hann window

$$STFTx[n](m,\omega) = X(m,\omega) = \sum_{n=0}^{N} x[n]w[n-m]e^{-j\omega n}, \tag{6}$$

where the length of the window is N and m indicates the window sampling rate.

The magnitude squared of the STFT yields the spectrogram of the function:

$$spectrogram\{x[n]\}(m,\omega) = |X(m,\omega)|^2. \tag{7}$$

We use a five-second window in our experiments. We show the spectrogram of six different motions of the y signal in Figure 4. The frequency is very small when ego repeatedly turns left and right. The 2 second period is almost the same as when ego repeatedly looks up and down. Looking up and down causes a frequency at 0.6 Hz. The spectrogram of small steps and walking are very similar. The largest frequency is at 7.8 Hz. When ego walks at 0.5 seconds per step, the frequency is 2 Hz. However, because the GoPro camera is head-mounted, the camera also has an oscillation when ego is walking. This camera oscillation causes this observed high frequencies. For jumping and running, the spectrogram gives accurate frequencies at 2Hz and 3.4 Hz, respectively.

We select the characteristic frequency of the window, which is defined as:

$$f_w = \begin{cases} f_{max}, & \text{if } f_{max} > 3\delta \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

where f_{max} is the frequency corresponding to the largest value in the spectrogram and δ is the standard deviation of the spectrogram. The condition of being larger than 3δ guarantees that the frequency picked is unlikely to be caused by noise.

In practice, we choose N to be 300 frames (5 seconds) and let the window moves 60 frames (1 second) each time. In this case, at each frame, there are 5 f_ws. We choose the median of these f_ws to be the final frequency at the frame.

We apply this procedure with the four movement signals x, y, r, and z and get four frequency signals f_x, f_y, f_r and f_z. In other words, in addition to four movement

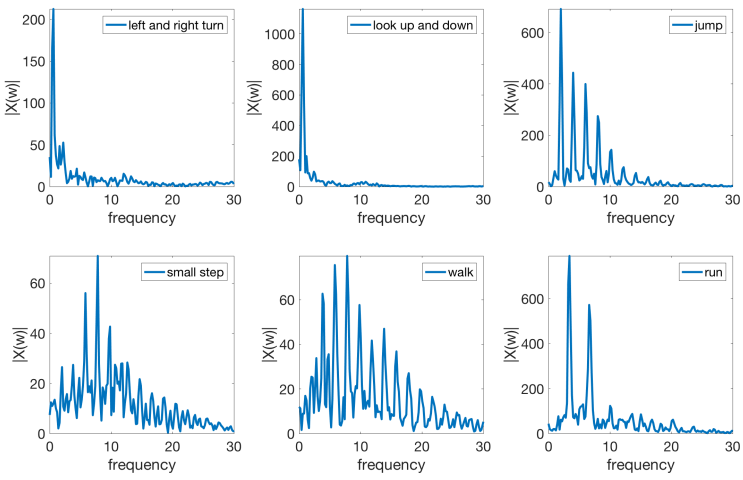

Fig. 4 Spectrogram of 6 kinds of motions

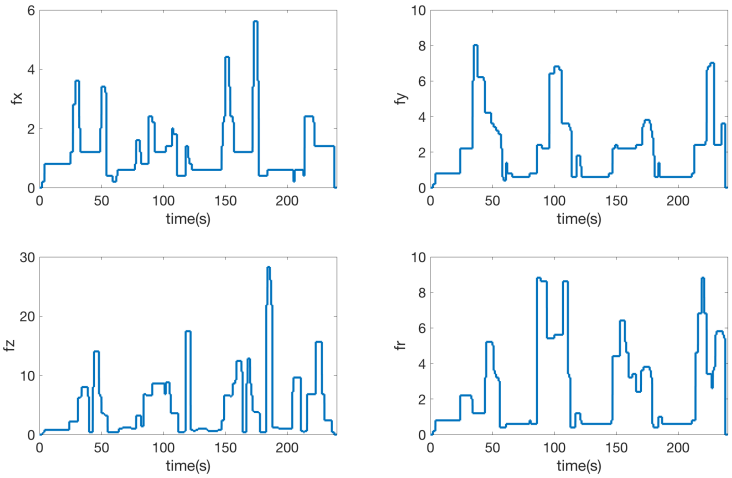

Fig. 5 The four characteristic frequencies fx, fy, fz, fr of the QUAD video.

signals, each frame transition is also associated with four characteristic frequencies. We compute these frequencies of the QUAD video and show their values in Figure 5. We can observe four periods in the frequencies which correlate with the action periods in the video.

2.4 Equalization of Variance

We always force the variance of each signal to be 1 by forcing x to be $\bar{x} + \frac{x-\bar{x}}{\sigma(x)}$, where $\bar{x}$ is the mean and $\sigma(x)$ is the standard variation. In this way, each signal gives equal contribution to the combined feature vector. Different weights can be considered to be applied on different signals based on the importance of the signals.

After equalizing the variance of the 8 signals, we combine them into a final motion feature f_{motion}. It is an $N \times 8$ matrix, where N is the number of frames in the video. Each row represents the eight-dimensional feature vector of one frame and we denote the feature vector of the ith frame to be F_i. In this way, we code the video frames by their feature matrix f_{motion}:

$$f_{motion} = [x, y, r, z, f_x, f_y, f_r, f_z]. \tag{9}$$

3 Classification Method

Once we have built the features f_{motion} of the video, we would like to infer a number of ego-motion categories from the data. In this section, we explore graph-based semi-supervised and unsupervised algorithms for video segmentation. We consider each transformation between two successive frames as a node in a weighted graph and classify them in different motion classes.

Recently, novel classification algorithms have been proposed [19] that are motivated by PDE-based image segmentation methods and are modified to apply to discrete data sets. These algorithms improve both accuracy of the solution and efficiency of the computation and can be potentially faster in parallel than various classification algorithms such as spectral clustering with K-means [35, 16]. The OpenMP parallelization and optimization of the algorithms are discussed in [18] with online demo and codes.

The novel classification algorithms consider each data point as a node in a weighted graph. The similarity (weight) between two nodes i and j is given by formula:

$$w_{ij} = exp(-||F_i - F_j||_2^2/\tau), \tag{10}$$

where F_i and F_j are feature vectors of nodes i and j according to (9), and τ is a parameter to be determined [7, 35]. We use the Euclidean distance here. To determine the value of τ, we try different values and run the experiments on the validation data

to choose the τ with the best accuracy. We use $\tau = 40$ in this paper. More about how to choose τ can be found in [4].

The classification problem is approached using ideas from graph-cuts [29]. Given a weighted undirected graph, the goal is to find the minimum cut (measured by a summation of the weights along the graph cut) for this problem. This is equivalent to assigning a scalar or vector value u_i to each i^{th} data point and minimizing the graph total variation (TV) $\sum_{ij}|u_i - u_j|w_{ij}$ [33]. Instead of directly solving a graph-TV minimization problem, the graph TV can be transformed to a graph-based Ginzburg-Laudau (GL) functional [4]:

$$E(u) = \varepsilon < L_s u, u > + \frac{1}{\varepsilon}\sum_i (W(u_i)) \tag{11}$$

where $W(u)$ is a double well potential, for example $W(u) = \frac{1}{4}(u^2-1)^2$ in a binary partitioning and multi-well potential in k dimensions (same as the number of classes). L_s is the normalized symmetric graph Laplacian which is defined as $L = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$, where D is a diagonal matrix with diagonal elements $d_i = \sum_{j\in V} w(i,j)$.

In the vanishing ε limit, GL recovers the graph TV functional [34]. Different fidelity terms are added to the GL functional for semi-supervised and unsupervised learning respectively. The GL energy for semi-supervised learning is:

$$E(u) = \varepsilon\langle L_s u, u\rangle + \frac{1}{\varepsilon}\sum_i W(u_i) + \sum_i \frac{\mu}{2}\lambda(x_i)||u_i - \hat{u}_i||^2_{L_2}. \tag{12}$$

The last term of Equation (12) is the regular L_2 fit to known data with some constant μ, while $\lambda(x)$ takes the value of 1 on fidelity nodes, and 0 otherwise. The variable $\hat{u}$ is the initial value for u with randomly chosen labels for non-fidelity data points and the "ground truth" for the fidelity points.

The GL energy for unsupervised learning is:

$$E(u, c_r) = \varepsilon\langle L_s u, u\rangle + \frac{1}{\varepsilon}\sum_i W(u_i) + \mu\sum_{r=1}^{\hat{n}}\langle ||f - c_r||^2, u_{\star,r}\rangle. \tag{13}$$

In (13), the term $||f - c_r||^2$ denotes an $N \times 1$ vector $(||f(x_1) - c_r||^2, ..., ||f(x_N) - c_r||^2)^T$ and the x_i $(i = 1, ...N)$ are the N pixels of the data set. In addition, the term $u_{\star,r}$ indicates the r^{th} column of u; the vector $u_{\star,r}$ is a $N \times 1$ vector which contains the probabilities of every node belonging to class r. The term $\hat{n}$ is the number of classes and is to be provided to the algorithm in advance. This problem is essentially equivalent to the K-means method when μ approaches $+\infty$.

The GL functional is minimized using gradient descent [15]. An alternative is to directly minimize the GL functional using the MBO scheme [21], or a direct compressed sensing method [20]. We use the MBO scheme in this paper in which one alternates between solving the heat (diffusion) equation for u and thresholding to maintain distinct class structure. Computation of the entire graph Laplacian is prohibitive for large data sets so we use the Nyström extension to randomly sample

the graph and compute a modest number of leading eigenvalues and eigenfunctions of the graph Laplacian [8]. By projecting all vectors onto this sub-eigenspace, the iteration step reduces to a simple coefficient update.

3.1 Semi-supervised and Unsupervised Algorithms

We outline here the semi-supervised and the unsupervised algorithms. For the semi-supervised algorithm, the fidelity data (a small amount of "ground truth") is known and the remaining data needs to be classified according to the categories of the fidelity. For the unsupervised algorithm, there is no prior knowledge of the data labels. We use the Nyström extension algorithm beforehand for both algorithms to calculate the eigenvalues and eigenvectors as the inputs. In practice, these two algorithms converge very fast and give accurate classification results.

Algorithm 2: Semi-supervised Graph MBO Algorithm [21]

Data: Eigenvectors matrix Φ, eigenvalues $\{\lambda_k\}_{k=1}^{M}$ and fidelity.
Result: u
Initialize u^0, $d^0 = \mathbf{0}$, $a^0 = \Phi^T \cdot u_0$;
while $\frac{||u^{n+1}-u^n||_2^2}{||u^{n+1}||_2^2} < \alpha = 0.0000001$ **do**
    a. Heat equation;
        1). $a_k^{n+1} = a_k^n \cdot (1 - dt \cdot \lambda_k) - dt \cdot d_k^n$;
        2). $y = \Phi \cdot a^{n+1}$;
        3). $d^{n+1} = \Phi^T \cdot \mu(y - u^0)$,;
    b. Thresholding;
        $u_i^{n+1} = e_r, r = \arg\max_j y_i$;
    c. Updating a;
        $a^{n+1} = \Phi^T \cdot u^{n+1}$
end

Algorithm 3: Unsupervised Graph MBO Algorithm [9]

Data: data matrix f, eigenvector matrix Φ, eigenvalues $\{\lambda_k\}_{k=1}^N$
Result: u

Initialize u^0, $a^0 = \Phi^T \cdot u^0$;
while $\frac{||u^{n+1}-u^n||_2^2}{||u^{n+1}||_2^2} < \alpha = 0.0000001$ **do**
    a. Updating c;
        $c_k^{n+1} = \frac{<f,u_k^{n+1}>}{\sum_{i=1}^N u_{ki}}$;
    b. Heat equation;
        1. $a_k^{n+\frac{1}{2}} = a_k^n \cdot (1 - dt \cdot \lambda_k)$;
        2. Calculating matrix P, where $P_{i,j} = ||f_i - c_j||_2^2$;
        3. $y = \Phi \cdot a_k^{n+\frac{1}{2}} - dt \cdot \mu P$;
    c. Thresholding;
        $u_i^{n+1} = e_r, r = \arg\max_j y_i$;
    d. Updating a;
        $a^{n+1} = \Phi^T \cdot u^{n+1}$;
end

The K-means algorithm [16] for finding K clusters proceeds iteratively by first choosing K centroids and then assigning each point to the cluster of the nearest centroid. The centroid of each cluster is then recalculated and the iterations continue until there is little change from one iteration to the next.

In both semi-supervised and unsupervised algorithms, we calculate the leading eigenvalues and eigenvectors of the graph Laplacian using the Nyström method [8] to accelerate the computation. This is achieved by calculating an eigendecomposition on a smaller system of size $M << N$ and then expanding the results back up to N dimensions. The computational complexity is almost $O(N)$. We can set $M << N$ without any significant decrease in the accuracy of the solution.

Suppose $Z = \{Z_k\}_{k=1}^N$ is the whole set of nodes on the graph. By randomly selecting a small subset X, we can partition Z as $Z = X \bigcup Y$, where X and Y are two disjoint sets, $X = \{Z_i\}_{i=1}^M$ and $Y = \{Z_j\}_{j=1}^{N-M}$ and $M << N$. The weight matrix W can be written as

$$W = \begin{bmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YY} \end{bmatrix},$$

where W_{XX} denotes the weights of nodes in set X, W_{XY} denotes the weights between set X and set Y, $W_{YX} = W_{XY}^T$ and W_{YY} denotes the weights of nodes in set Y. It can be shown that the large matrix W_{YY} can be approximated by $W_{YY} \approx W_{YX} W_{XX}^{-1} W_{XY}$, and the error is determined by how many of the rows of W_{XY} span the rows of W_{YY}. We only need to compute W_{XX}, $W_{XY} = W_{YX}^T$, and it requires only $(|X| \cdot (|X| + |Y|))$ computations versus $(|X| + |Y|)^2$ when the whole matrix is used.

4 Experimental Results

To evaluate the performance of our method we need both choreographed video sequences to run controlled experiments and real-world videos to observe performance of our method in naturalistic settings. It is easy to define the ground truth for the choreographed videos since the motions of the person who takes the video are both discrete, and well-defined. For example, looking left and right never coincides with running. However, real-world body-worn video usually contains a combination of different motions with noise and it is therefore harder to define a ground truth.

4.1 Choreographed Video

The first video we use is QUAD [13]. We show one frame of the QUAD video in Figure 6. This video is 4 minutes and 10 seconds in length and has 60 frames per second. It contains 9 ego-motions (stand still, turn left, turn right, look up, look down, jump, step in place, walk and run). Ego used a head-mounted GoPro camera. Ego performed the 9 actions in order and repeated them four times. The ground truth is shown in the first row of Figure 7. The horizontal axis represents time and colors represent different ego-motion categories. The order of the movements are standing still, turning left and turning right repeatedly, looking up and looking down repeatedly, jumping, stepping, walking, running, turning left and then start the same series of motions again for another three times. We compute the feature vector for each

Fig. 6 One frame of the QUAD video.

two successive frames as described in section 2. Then we use K-means, the unsupervised graph MBO algorithm and the semi-supervised graph MBO algorithm for the ego-motion classification. We use 10% known labels (evenly sampled) in the semi-supervised graph MBO algorithm. The classification results of these three al-

gorithms are shown in the 2nd, 3rd and 4th rows of Figure 7. For the K-means and the unsupervised MBO algorithm, we ran the experiments several times and pick the best results here. Depending on the initialization, these two algorithms can converge to different local minima, which is common for most non-convex variational methods. The K-means algorithm gives relatively good results, except that it does not recognize the category of looking down and misclassifies some parts of running, jumping, small steps and walking. The unsupervised graph MBO algorithm gives results similar to K-means. The semi-supervised graph MBO algorithm with 10% known labels gives very accurate results. The accuracy summary of these three algorithms is shown in Table 2.

Table 2 Accuracy Summary of the QUAD data set

Accuracy	Overall	Average	1.Stand still	2.Turn left	3.Turn right	4.Look up
K-means	**64.84%**	**61.79%**	95.82%	72.26%	77.28%	73.24%
Unsupervised MBO	**66.62%**	**67.59%**	79.99%	76.82%	83.37%	69.41%
Semi-supervised MBO	**89.14%**	**88.74%**	87.90%	89.43%	92.80%	80.36%

Accuracy	5.Look down	6.Jump	7.Step	8.Walk	9.Run
K-means	0	83.29%	49.29%	36.66%	68.25%
Unsupervised MBO	77.82%	39.38%	43.54%	83.27%	54.68%
Semi-supervised MBO	84.59%	92.71%	93.98%	84.52%	92.38%

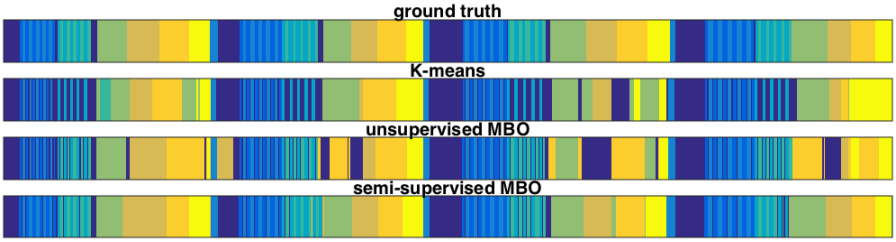

Fig. 7 Ego-motion classification results of the QUAD video. The 9 colors represent 9 different ego-motion classes: standing still (dark blue), turning left (moderate blue), turning right (light blue), looking up (dark green) and looking down (light green), jumping (bud green), stepping (aztec gold), walking (orange), runing (yellow).

4.2 Real-world Body-worn Video

We also investigated real-world body-worn videos. We use a data set from the Los Angeles Police Department. The videos are from police wearing chest-mounted

cameras while patrolling areas of Los Angeles on foot. The videos record a wide array of police activities from basic patrol through foot chases and arrest. Our ego-motion classification results may be used in modeling the routine activities of police and their interactions with the public.

Police BWV is not collected under controlled circumstances. Ego-motions may evolve rapidly without clear or discrete transitions. Much body worn video is collected at night impacting light and color saturation. The videos also have distortion due to the use of a fish-eye lens. Since there has been very little formal analysis of police BWV, there is a lack of appreciation for the diversity of police behavior likely to be encountered (i.e., very limited semantic dictionaries). The ground-truth is labeled by us without input from the police.

We show here the video segmentation result of one clip of police video. The video is 8 minutes and 16 seconds in length, with 14991 frames in total. In the video, police arrive at an apartment building, talk with some people in front of the building, go upstairs, wait outside a room, enter and search the room, leave the room, walk downstairs, and talk to several people outside the building. We define four ego-motion categories in this video – standing (or very slow motions not easy to define), walking, going upstairs, and going downstairs. The ground truth classification of this video is shown in the first row of Figure 8. The dark blue segments represent the category of standing or slow movements when the officer talks with others in front of the building. It also contains actions when the officer enters the room. The video of this period is very shaky and not easily defined as one motion category. The light blue segment corresponds to the walking category. The green segment corresponds to the police going upstairs and the yellow part is going downstairs.

We explore the same algorithms for the police body-worn video. We are not using the unsupervised graph MBO algorithm because the result is not consistent. The results are shown in Figure 8. K-means captures the difference between going upstairs and downstairs. However, K-means frequently misclassifies walking and going downstairs. Some standing frames are classified as other motion categories. This later result is reasonable since standing in this video combines some other movements. Then we use the semi-supervised graph MBO algorithm with 10% known labels on this piece of video. The segmentation results are shown in the third row of Figure 8. It can be seen that the result is much better than K-means, and the four categories are all captured almost correctly. The accuracy summary is shown in Table 3. The overall accuracy of the semi-supervised graph MBO algorithm with 10% known labels is 90.17%.

Table 3 Accuracy Summary of the police body-worn video data set

Accuracy	Overall	Average	1.Stand	2.Walk	3.Upstairs	4.Downstairs
K-means	**63.62%**	**63.77%**	68.91%	37.78%	91.84%	56.53%
Semi-supervised MBO	**90.17**%	**74.09%**	96.10%	82.12%	83.45%	34.71%

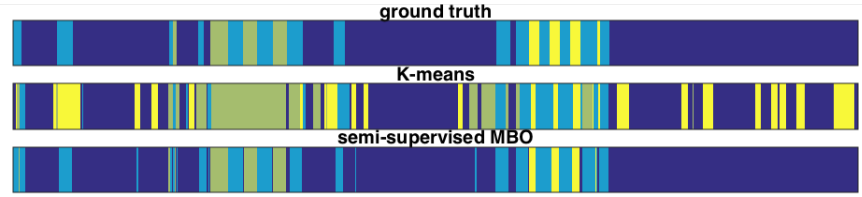

Fig. 8 Ego-motion classification results of the police video. The 4 colors represent 4 different ego-motion classes: standing or very slow motions and motions not easy to define (dark blue), walking (light blue), going upstairs (green) and going downstairs (yellow).

5 Conclusion

In this paper, we investigate the task of discovering ego-motion categories from first-person videos. We deal with this problem in two steps. The first step is comparing two successive frames using the inverse compositional algorithm to extract signals containing motion and motion frequency information. Then we use unsupervised and semi-supervised clustering algorithms for classification. The semi-supervised graph based methods are particularly accurate using only 10% training data. We show promising results on both choreographed and real-world video data.

The potential for future advances in this area are significant particularly in relation to police body-worn video. At full deployment of body-worn video in 2018, the Los Angeles Police Department is projected to collect 3.2 million individual videos totaling more than 200K hours of total video feed per year. This represents both a vast resource and a significant analytical challenge. The amount of data suggests that the full array of ego-motions practiced by police might eventually be discovered and subject to classification, moving us towards a realistic picture of the diversity of police activities. There will clearly be no lack of training data with which to tackle this problem. The same surfeit of video data is proving to be true in other domains outside of policing. Recognition of the diversity of ego-motion in policing activity may also lead to novel extensions of the methods into dyadic- and n-person motion models. In the dyadic-motion case there is much to be learned. It is well known that relative motion of individuals with respect to one another encodes fundamental social information [3]. For example, an individual running away from ego may encode avoidance or fear, while an individual running directly towards ego may encode attraction and threat. More complex social interactions may be captured in n-person motion models.

The challenges to achieving such outcomes with real-world video are also significant. In the police body-worn video case, semi-supervised classification clearly outperforms the unsupervised approach. Yet even a small fraction of fidelity points (10% in the current method) is probably infeasible given the volumes of video arriving each day. Semi-supervised methods will therefore need to rely on as few fidelity points as possible. However another approach is video labeling where activities segmented in one video might be used as labels for semi-supervised segmentation in

another video. This was demonstrated in [4, 5] for image labelling. It will also be necessary to consider how generalizable methods are across real-world video examples. Ideally, a handful of videos might be exhaustively labeled for ground-truth and these would then work across the growing set of videos. This is an empirical questions that we can start addressing now with the recognition that new methods may be needed to account for the variability of real-world video.

Finally, we also point out that body-worn video is but one sensor platform in what is increasingly a multi-sensor world. It is worth investigating whether there is an advantage to doing more with single sensors, or whether it is better to integrate the signals from many independent sensors. For example, we can imagine doing both ego-motion and scene topic classification from the same video sequence, or as an alternative use accelerometers to capture ego-motion and matching these data to scene classification from video. Importantly, the issues are not strictly technological. Police body-worn video is treated as evidence and therefore is subject to all of the evidence handling rules required by law. Each sensor implies a different packet of physical of evidence that must be maintained and handled appropriately. Future work will need to examine these sorts of tradeoffs in detail.

6 Acknowledgements

The work was supported by the ONR grant N00014-16-1-2119, NSF grant DMS-1737770, NSF grant DMS-1417674, FUI project Plein Phare by BPI-France and NIJ Grant 2014-R2-CX-0101.

References

1. S. Baker and I. matthews, Lucas-kanade 20 years on: A unifying framework, International journal of computer vision, pp. 221–255, 2004
2. S. Baker and I. Matthews, Equivalence and efficiency of image alignment algorithms, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001
3. H. C. Barrett, P. M. Todd, G. F. Miller and P. W. Blythe. Accurate judgments of intention from motion cues alone: A cross-cultural study, Evolution and Human Behavior vol. 26, no. 4, pp. 313-331, 2005.
4. A.L. Bertozzi and A. Flenner: Diffuse interface models on graphs for classification of high dimensional data. Multiscale Modeling & Simulation 10.3: 1090-1118 (2012)
5. A. L. Bertozzi and A. Flenner, Diffuse Interface Models on Graphs for Classification of High Dimensional Data, SIAM Review, 58(2), pp. 293-328, 2016.
6. P. Bouthemy, M. Gelgon and F. Ganansia, A unified approach to shot change detection and camera motion characterization, IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 7, pp. 1030-1044, 1999
7. F. Chung: Spectral graph theory. Vol. 92. American Mathematical Soc., (1997)
8. C. Fowlkes, S. Belongie, F. Chung, J. Malik: Spectral grouping using the Nyström method. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26.2: 214-225 (2004)

9. H. Hu, J. Sunu, A.L. Bertozzi: Multi-class graph Mumford-Shah model for plume detection using the MBO scheme. Energy Minimization Methods in Computer Vision and Pattern Recognition. Springer International Publishing (2015)
10. T. Huynh, M. Fritz and B. Schiele, Discovery of activity patterns using topic models, Proceedings of the 10th international conference on Ubiquitous computing, pp. 10-19, 2008
11. V. Kiani and H. R. Pourreza, Robust GME in encoded mpeg video, Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia, pp. 147-154, 2011
12. J. Kim, H. S. Chang, J. Kim, and H. Kim, Efficient camera motion characterization for MPEG video indexing, 2000 IEEE International Conference on Multimedia and Expo, vol. 2, pp. 1171-1174
13. K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, Fast unsupervised ego-action learning for first-person sports videos, Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 3241-3248, 2011
14. B. D. Lucas and T. Kanade: An iterative image registration technique with an application to stereo vision. Proceedings of the 7th International Joint Conference on Artificial intelligence (IJCAI), 1981.
15. X. Luo and A.L. Bertozzi: Convergence analysis of the graph Allen-Cahn scheme. Preprint.
16. J. MacQueen: Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967.
17. W. W. Mayol and D. W. Murray, Wearable hand activity recognition for event summarization, International Symposium on Searable Computers, pp. 122-129, 2005
18. Z. Meng, A. Koniges, H. Yun, S. Williams, T. Kurth, B. Cook, J. Deslippe and A.L. Bertozzi: OpenMP Parallelization and Optimization of Graph-Based Machine Learning Algorithms. In: Maruyama N., de Supinski B., Wahib M. (eds) OpenMP: Memory, Devices, and Tasks. Lecture Notes in Computer Science, vol 9903. Springer. IWOMP 2016.
19. Z. Meng, E. Merkurjev, A. Koniges and A.L. Bertozzi: Hyperspectral Image Classification Using Graph Clustering Methods. Image Processing On Line, 2017
20. E. Merkurjev, E. Bae, A.L. Bertozzi and X.C. Tai: Global binary optimization on graphs for classification of high-dimensional data. Journal of Mathematical Imaging and Vision, 52(3), 414-435.
21. E. Merkurjev, T. Kostic, A.L. Bertozzi: An MBO scheme on graphs for classification and image processing. SIAM Journal on Imaging Sciences 6.4: 1903-1930 (2013)
22. H. Pirsiavash and D. Ramanan, Detecting activities of daily living in first-person camera views, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2847-2854, 2012
23. X. Ren and C. Gu, Figure-ground segmentation improves handled object recognition in egocentric video, CVPR, vol. 2, no. 3, pp. 6, 2010
24. J. Sánchez, The Inverse Compositional Algorithm for Parametric Registration, Image Processing On Line, pp. 212-232, 2016
25. J. Sánchez and J. Morel, Motion smoothing strategies for video stabilization
26. E. H. Spriggs, F. de la Torre and M. Hebert, Temporal segmentation and activity classification from first-person sensing, 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 17-24, 2009
27. T. Starner, B. Schiele, and A. Pentland, International Symposium on Wearable Computers, pp. 50-57,1998
28. T. Starner, J. Weaver and A. Pentland, Real-time american sign language recognition using desk and wearable computer based video, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 12, pp. 1371-1375, 1998
29. M. Stoer and F. Wagner: A simple min-cut algorithm. Journal of the ACM (JACM) 44.4 : 585-591 (1997)
30. L. Sun, U. Klank and M. Beetz, EYEWATCHME3D hand and object tracking for inside out activity analysis, 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 9-16, 2009

31. S. Sundaram and W. Cuevas, High level activity recognition using low resolution wearable vision, 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 25-32, 2009
32. R. Szeliski: Computer vision: algorithms and applications. Springer Science & Business Media, 2010.
33. A. Szlam and X. Bresson: A total variation-based graph clustering algorithm for cheeger ratio cuts. UCLA CAM Report : 09-68 (2009)
34. Y. Van Gennip and A. L. Bertozzi: *Gamma*-convergence of graph Ginzburg-Landau functionals. Advances in Differential Equations 17.11/12: 1115-1180 (2012)
35. U. Von Luxburg: A tutorial on spectral clustering. Statistics and computing 17.4: 395-416 (2007)