Distributed Adaptive Model Predictive Control of a Cluster of Autonomous and Context-Sensitive Body Cameras

Shiwei Fang
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC, USA
shiwei@cs.unc.edu

Ketan Mayer-Patel
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC, USA
kmp@cs.unc.edu

Shahriar Nirjon
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC, USA
nirjon@cs.unc.edu

ABSTRACT

Increasing deployment of body cameras by the law enforcement agencies makes us rethink the relation between the camera and the public. In contrast to current implementations of a body camera that use a power-hungry default configuration and can only be turned on and off by an officer, we propose an idea that the camera should be autonomous and active all the time. By leveraging the information from an on-board inertial measurement unit (IMU), these autonomous cameras should dynamically adjust their configuration in order to keep the device under the desired energy budget. To enable such a system, we propose a distributed adaptive model predictive controller for a system of body cameras, which allows the collaboration between multiple cameras which is currently not available in existing implementations.

Keywords

Model Predictive Control; Context-Sensitive; Body Camera

1. INTRODUCTION

In recent years, we have witnessed some upsetting conflicts between the law enforcement personnel and the public. Among many solutions proposed to deal with this problem is the use of body cameras [8]. A body camera is a wearable device that records video during an event and provides evidence about what happened during future investigations. However, the current policy lets the law enforcement agencies determine whether or not to release those footages, and thus create an imbalance in transparency and power. In this paper, we envision a body camera that is autonomous and always active, whose control is not in the hands of the law enforcement officers, and the public has the same right to the recorded videos. We hope that such a device will bring the intended mutual trust between the law enforcement agencies and the communities they serve.

Existing implementations of body cameras typically consist of an on/off switch that is controlled by its wearer. This design suits the purpose in an ideal scenario but not in many real-world situations as the officer cannot predict when to turn on the camera and may forget to do so. There are some implementations which trigger the camera on when a gun is pulled within a certain distance, however, it is still missing the information as for how the situation has developed. We argue that the body camera needs to be autonomous and active at all time. However, due to the constraints of the system such as battery life, processing power, and storage size, the system needs to be adjusted accordingly. When there is no event, the body camera should be in a low power mode which will only record information in case of an emergency. This configuration will allow the camera to record as much information as possible, without requiring significant modifications to it or a bulky battery pack. We envision the body camera should be as small as possible thus does not affect the officer in any aspect.

Furthermore, current body cameras do not support collaborations when there are multiple cameras present – which is usually the case where more than one officers respond to the same scene. These situations require more attention as they are usually more chaotic. We propose that by leveraging the system configuration, pose, and position of each camera, the scene can be recorded more efficiently. In contrast to a centralized system, our controller does not depend on other cameras which may become inoperable and lose information. The proposed controller guarantees the camera records at a certain standard and improves the performance in a cluster of cameras. We make the following contributions in this work:

- We design and implement a low-cost, open-source body camera prototype. We identify the system variables that are sensitive to output power and video quality and model those variables.
- We propose an adaptive model predictive controller that optimizes the system in terms of power consumption and video quality to guarantee the desired system lifetime.
- We perform a simulation-based evaluation of the controller in both single camera and collaborative scenarios. Results show that the lifetime of the proposed system is 33%-40% more than the baseline.

WearSys'17, June 19, 2017, Niagara Falls, NY, USA
© 2017 ACM. ISBN 978-1-4503-4959-8/17/06...$15.00
DOI: http://dx.doi.org/10.1145/3089351.3089358

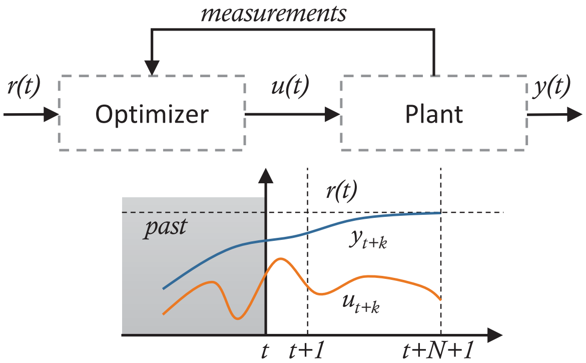

Figure 1: Model predictive controller

2. BACKGROUND

2.1 Context Recognition

Human Activity Recognition (HAR) has being studied extensively and numerous methods have been proposed. There are challenges in this area such as *intraclass variability*, *interclass similarity*, and *null class problem* [3]. To solve these problems, researchers have proposed to use different sensors like IMU, GPS, and camera. The method for data segmentation, which is required for recognition at a certain time point such as sliding window, energy based, and additional sensors [3] [1], is crucial to the classification accuracy. After the data is processed, a classifier like Dynamic Time Warping (DTM) [2], Hidden Markov Models(HMMs) [4] or kNN [9] [5] is used for training and classification. These steps are referred as activity recognition chain and a more detailed review can be found in [3].

2.2 Feedback Control

A feedback controller is to observe the output signals of the unit under control and to compute and apply the right input signal to the unit. A standard feedback controller like PID [6] controller requires an accurate model of the system which is extremely difficult to acquire due to the presence of *disturbance* as the system operates in an open, real-world environment, especially for the ones that involve human. *Adaptive controllers* [10] are a special kind of feedback controllers which can update the model to better describe the system.

A *Model Predictive Controller* (MPC) [12] uses a dynamic model of the plant to predict its future outputs and optimizes the control signals by solving an optimal control problem over a finite future horizon. Figure 1 shows a model predictive controller and its inputs and outputs over time. The *plant* in the figure is the unit being controlled. In the figure, $y(t)$ is the output of the system, $r(t)$ is the reference output, and $u(t)$ is the control input. The shaded area on the left denotes the past and the area on its right $(t, t+N)$ is the finite prediction horizon. At each time point t, predicted outputs and manipulated inputs are computed by the controller over the finite time horizon $t+0, \ldots, t+N$, and only the first input $u(t)$ is applied to the system.

Model predictive controllers have several advantages such as explicitly handle constraints on inputs and outputs, and the performance is optimized by solving an open-loop optimization problem. These controllers are widely used in different industrial application areas such as air and gas, chemicals, and food processing [11, 7].

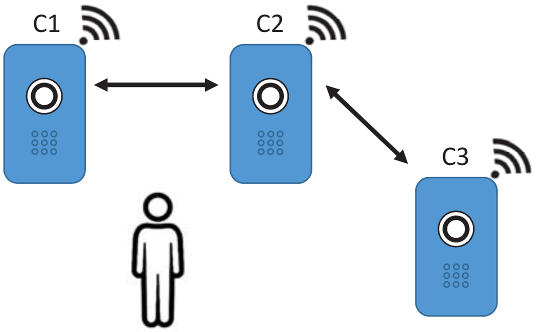

Figure 2: System overview. C1, C2, and C3 are body cameras. Not drawn to scale.

3. DESIGN CONSIDERATIONS

3.1 Overview

Body cameras can present in various environments where it can work alone or in collaboration with others. In an environment where there is only one camera present, the controller only considers the context it is in. Based on the measurements from the inertial measurement unit (IMU), battery status and internal analysis, the controller will change the camera configuration to maximize the video quality while keeping the system within the intended energy budget. This allows the camera being turned on all the time but not consumes many resources whether its battery life or storage space.

In a collaborative fashion where multiple body cameras are present as shown in Figure 2, camera C1 and C2 are recording the scene from the same angle while camera C3 is recording from the side. The different system status and position of each camera can be used to incorporate collaboration between multiple cameras. The controller in each body camera will take into account of other cameras' presence, however, the configuration does not rely on others which may go offline at any point in time. For instance, camera C1, with an appropriate amount of battery remaining, won't stop recording or record at a very low quality even if it learns that camera C2 and C3 both are recording at the highest quality and have more battery remaining.

3.2 System Goals

The goal of this system is to have a body camera controller that can adjust the configuration according to the environment it is in, thus allows the body camera to be always active but does not consume excessive energy or only recording at a low quality.

3.2.1 User Context

The quality of the video is based on the current context that the officer is in. The context is determined based on the state machine and IMU measurements. We define the set of contexts as *in-car*, *walking*, *running*, and *confronting*. If the officer is sitting in a car, there is no need to record at a higher frame rate or a better quality as the scene is mostly static. However, in a confronting context which can result in violence, the camera needs to preserve the maximum information thus requires a high-quality video that is suitable for the context. The different contexts form a state machine as shown in Figure 3 which we use to track which state the camera is currently in.

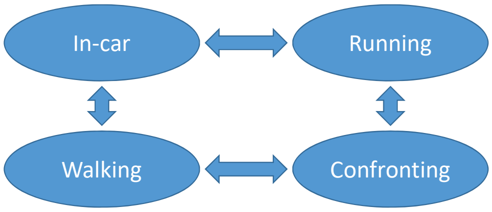

Figure 3: State diagram

3.2.2 Collaboration of Camera Cluster

During an intensive situation, multiple body cameras may present. We argue that the information preserved would not be maximized if all cameras are set to the same configuration. Consider a situation where multiple cameras are facing the same direction, some cameras can record at a higher resolution but a lower frame rate to get a clear view of the scene and other cameras can record at a lower resolution with a higher frame rate to get the detailed action sequence of the event. The frame rate and resolution have a lower bound in this situation which allows for guaranteed video quality when cameras are treated individually, thus the controller does not depend on others. However, if the battery is low on the camera, the resolution and frame rate will be further dropped automatically as recording something is better than nothing. In Figure 2, camera C1 and C2 can use complementary resolution and frame rate settings, as for C3, it will choose the configuration which can find the balance between quality and power consumption.

The cameras will also help each other to save power. In Figure 2, camera C1 has more battery remaining than camera C2, in this situation, camera C2 can choose a lower video quality configuration to save power while camera C1 recording at original quality. The idea behind this capability is that slightly lower quality video does not significantly degrade the viewer's experience and there are alternative footages with better quality available, however, a longer battery life is more important and crucial to the system.

3.2.3 Adaptive Model Predictive Control

The body camera is resource constraint device which requires us to minimize the utilization and lower the power consumption when possible. The controller controls the parameters of the camera with limited CPU usage to minimize the power consumption. The time horizon of the controller is adaptive due to the fact that different officers have different behaviors. For instance, one officer normally spends 15 minutes in confronting context and another officer for 30 minutes. Furthermore, the system model requires updates for any inaccurate modeling and refinement for each single unit. The adaptive natural of the controller enables a more accurate control and can improve system performance over time.

The controllable parameters we are interested in are resolution, frame rate, camera sensor mode, and controller activation interval. The parameters are set according to current context and power requirements. The predefined system model and cost function will optimize the configuration for our objectives while remaining in the constraints. For instance, if the current context is *in-car*, the controller will lower the frame rate and resolution to reserve power. The configurations and current system status are in a dynamic relation which a heuristic approach is not capable of handling all situations.

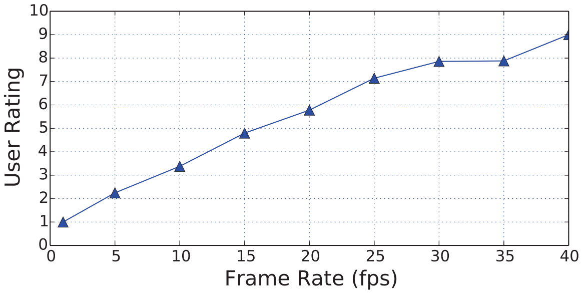

Figure 4: User Rating vs. Frame Rate

4. CONTEXT RECOGNITION

Context recognition is a crucial component of our controller. We use an IMU sensor to perform human activity recognition for four types of activities: *running*, *walking*, *sitting*, and *standing*. The detected activity is used in transitions between different states in the state machine of Figure 3. We choose IMU's 3-axis accelerometer data to extract features, which we found to be more accurate in activity detection than using a combination of accelerometer with gyroscope and magnetometer. The means and variances of each axis are extracted for a time window. Inspired by the work in [1], multiple time windows are combined to create more features. In our implementation, we choose 0.7 and 1.4s time windows, which give us a total of 18 features. To balance the constrained nature of the body camera and accuracy requirements, we use a kNN [5] classifier. Based on initial cross-validation experiments, we set k = 9 to ensure a stable performance which achieved an accuracy of over 90%. A set of three consecutive classification results, $\{x_i\}$, is used for majority voting to detect an activity.

5. DESIGN OF CONTROLLER

In this section, we discuss the design of the proposed body-cam controller. We name the scenario where a single camera is present as *local environment* and a cluster of cameras as *collaboration environment.*

5.1 Local Environment

In a local environment, the controller only takes into account the internal control inputs, which are the current camera status, context, and battery level. To define the objective function of the controller that maintains the desired quality level of captured videos, at first, we conduct a user study on how human users rate the quality of a video when its resolution and frame rate are varied. We record a video at a high quality and convert it into multiple videos with different configurations. At first, we ask the users to rate the videos which have the same resolution but different frame rates. The mean value of the normalized score for each frame rate is shown in Figure 4. We find that it becomes harder for an average user to notice the difference when the frame rate exceeds 25 frames per second. We then ask the users to rate the videos which have the same frame rate but are of different resolutions. The result is shown in Figure 5. From the result, we find that the users can easily tell the difference and prefer a higher resolution. Based on this study, we assign each camera configuration (i.e. resolution and frame rate) a score that represents human perception quality. The score is used in the objective function of our controller.

For the controller, we define its objective as to match the

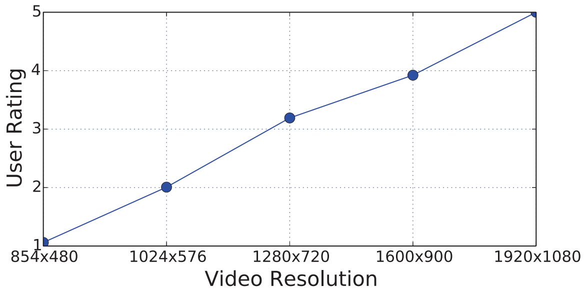

Figure 5: User Rating vs. Video Resolution

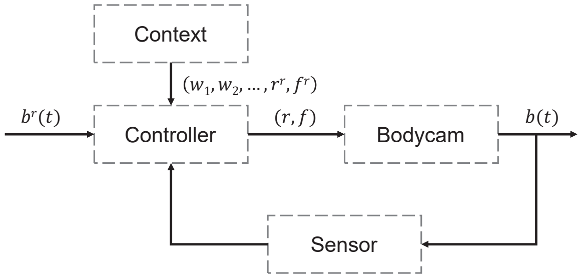

Figure 6: Controller design for the body camera

actual power consumption to a reference value (e.g., set by policy or learned during deployment) while maintaining a certain quality level. For instance, if the targeted battery life of the system is 8 hours and the camera with a single configuration will drain its battery linearly during that period, the camera system should have 50% battery remaining when it's 4 hours into the day. In that case, the linearly dropping battery level will be the reference at any point in time. We define the cost function to optimize as:

$$J = (b - b^r)^2 + \sum_{i=0}^{N} w_i S(x_i - x_i^r) \tag{1}$$

where

$$S(x) = \frac{1}{1 + e^{-a \cdot x}} - 0.5 \tag{2}$$

In Equation 1, b is the future battery level after applying the camera setting for the time step, b^r is the reference battery level, x_i and x_i^r are the controlled variable and the reference for controlled variable, w_i are the weights of these variables, S is the sigmoid function shown in Equation 2, where a is tuned for the desired controller property on what range of the camera configuration we want to focus on and where to allow for more aggressive control. The sigmoid terms are used to reward or penalize certain behavior in each context, and each context has its own reference values for control variables and weights. The cost function can be extended by adding more control variables as sigmoid terms. Sigmoid terms are used for it's decreased rate of change when the actual value is away from the reference value, which enables the system to take a more aggressive action when necessary.

We design the body camera controller system as shown in Figure 6. The controller takes input from the sensor which measures the system's battery level $b(t)$. The context state machine provides objective function's weights $(w_1, w_2, \ldots)$ and reference values (as x_i^r in Equation 1) of resolution r^r and frame rate f^r for the current context. The controller computes the control input (r, f) (as x_i in Equation 1) based on these parameters and the reference value of the desired battery level $b^r(t)$.

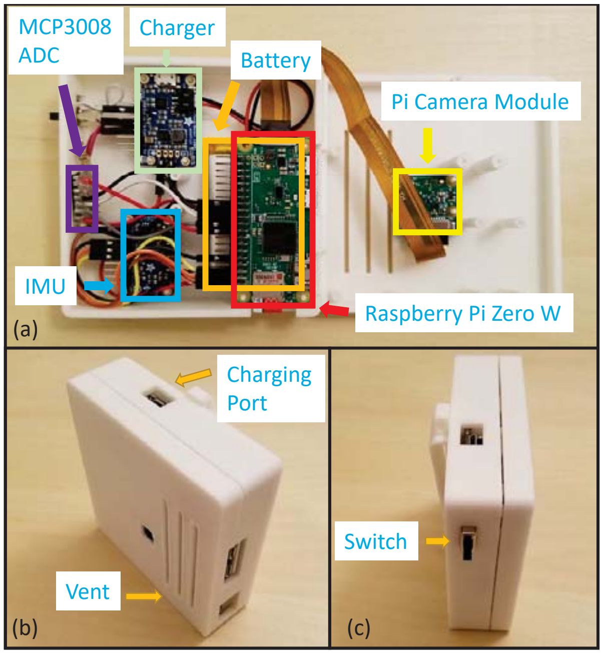

Figure 7: Body camera. (a) All components are shown. Battery is under Raspberry Pi Zero W. (b) Right-front view. Camera near the vent. (c) Left view. Switch, charging port, and clip.

The model predictive controller needs to optimize over a finite time horizon. The horizon is set to the next five likely contexts and their durations, given the current context. The sequence of contexts and their durations are adaptive and refined for each bodycam over time. The battery power consumption for each camera configuration is updated to minimize the impact of any inaccuracy in system modeling and changes in hardware performance such as battery degrading.

5.2 Collaboration Environment

For a collaboration environment, the weights in Equation 1 are used to reflect the difference between multiple cameras. For instance, if camera A has more remaining battery than B, B's controller chooses a less power hungry configuration. We assume the communication system is already available.

6. SYSTEM IMPLEMENTATION

6.1 Hardware Design

We have implemented a prototype of the proposed body camera which is shown in Figure 7. We use a Raspberry Pi Zero W and a Pi Camera Module V2 as the main components. We connect an IMU to the Raspberry Pi via I^2C. The IMU has LSM9DS0 chip on-board with a 3-axis accelerometer, a 3-axis magnetometer, and a 3-axis gyroscope.

We use a PowerBoost 1000C power supply to charge the battery and supply 5V to the Raspberry Pi. For the battery monitoring, we use a MCP3008 ADC chip to measure the battery voltage and convert that to a percentage level used by the controller. For this conversion, we use an empirically estimated model. We measure the voltage over time for four batteries of the same model while draining them with

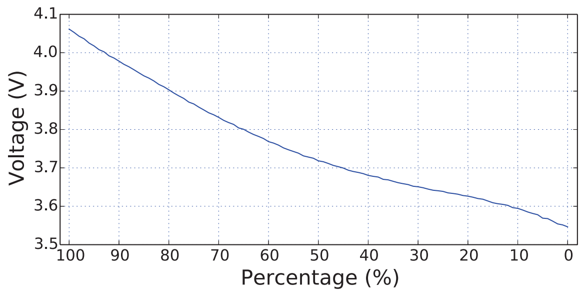

Figure 8: Battery Voltage vs. Remaining Percentage

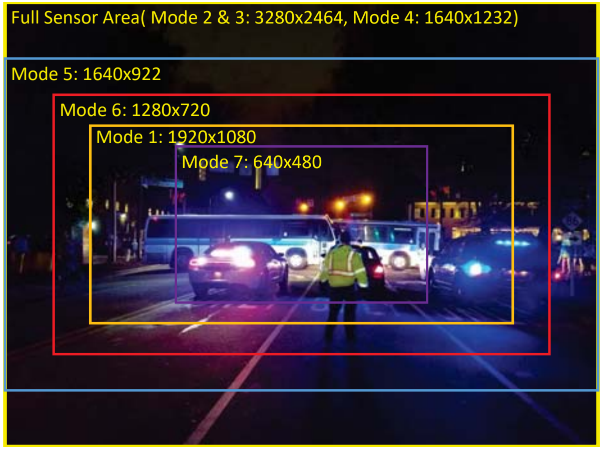

Figure 9: Field of view for each sensor setting

a constant current. The relation between the voltage and remaining percentage is shown in Figure 8.

6.2 Software Design

We use the *picamera* Python library to control the camera configuration and *scikit-learn* Python library for kNN [5] classification. We choose 0.1s as the interval to collect IMU data which can acquire sufficient information while maintain minimal CPU utilization. The battery voltage is queried every 10 seconds and the percentage is calculated from the average voltage over 30 seconds.

7. EVALUATION

7.1 System Measurement

Our system identification model is generated through the measurements of different settings performed in this section. We measure the power consumptions of the Raspberry Pi Zero W and Raspberry Pi Camera Module V2. The Pi Camera module sensor has 7 modes in total, which are shown in Figure 9. In this paper, we use mode 2 and 4, which utilize the full sensor area and have the same field of view. For mode 2, the frame per second rate varies from 0.1 to 15, and for mode 4 it is varied from 0.1 to 40.

We measure the power consumption of the system for different resolutions and frame rates. Due to constraints of the Raspberry Pi platform, the maximum frame rate for 1920x1080 resolution is 30 frames per second, however, the camera module is capable of recording at 40 frames per second in mode 4.

Figure 10 shows the power consumption measurements of the system when we use sensor mode 4. In this mode, the

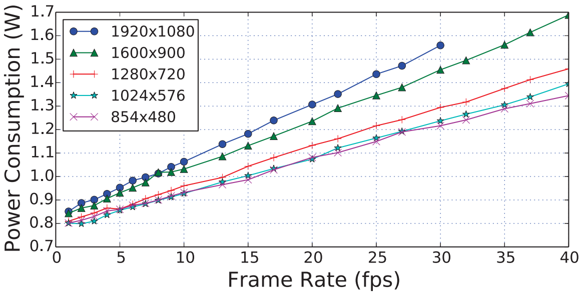

Figure 10: Power consumption with sensor mode 4

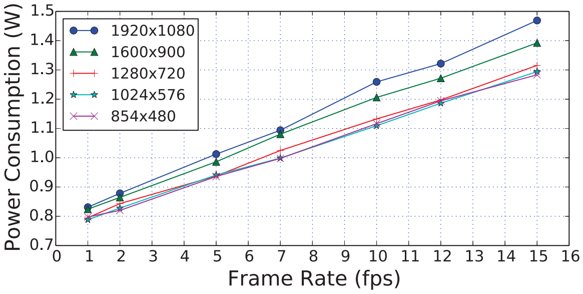

Figure 11: Power consumption with sensor mode 2

native resolution of the sensor is 1640x922. As a result, the common resolutions we choose in this paper requires the system to re-size the video. For 1080p resolution, the system sizes up the resolution, which degrades the video quality in contrast to down-size the video from a higher resolution captured from the camera. As shown in Figure 11, the power consumption is higher in mode 2 than it is in mode 4 due to the use of the full sensor which requires more power, and so is resizing and encoding. Due to the downsizing of the video to 1080p, the video quality is higher than it is in sensor mode 4.

We also measure the power consumption of the system under different CPU utilizations. The measurements are shown in Figure 12. As expected, when the camera is running at a higher resolution and frame rate, the CPU utilization increases.

7.2 Algorithm Evaluation

We conduct simulation-based experiments (based on real data and real measurements) to evaluate the performance of the body camera controller for both local and collaboration environment.

7.2.1 Local Environment

We design a sequence of contexts for this environment: in-car (0.5), walking (0.1), confronting (0.5), walking (0.1), in-car (0.6), running (0.1), confronting (1), walking (0.2), and in-car (0.5), where the number in the parentheses is the duration in hours. Using our pre-defined weights, the simulation results are shown in Figure 13. The reference model is a configuration which lasts for 8 hours on a fully charged 9.6Wh battery. The referenced 1.2 W power consumption corresponds to 1080p at 15 fps, 900p at 18 fps, 720p at 24 fps, 1024x576 at 28 fps, and 480p at 29 fps. These configurations are not of good quality. The baseline model is the camera running at 900p at 40 fps. Our controller chooses the same setting as the baseline model for confronting and low frame rate for in-car. Our method has a remaining battery

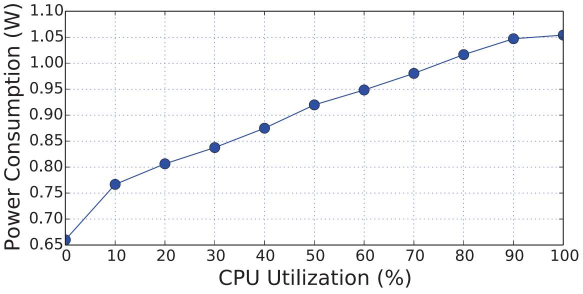

Figure 12: Power Consumption vs. CPU Utilization

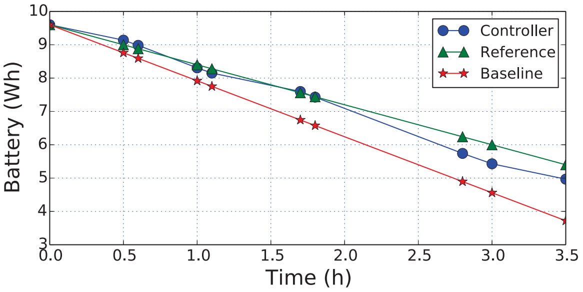

Figure 13: Simulation of the controller compared to baseline implementation

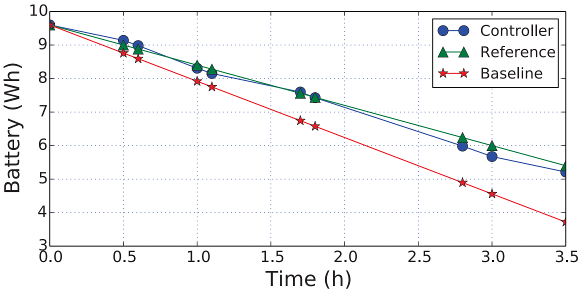

Figure 14: Simulation of the controller with collaboration compared to baseline implementation

of 4.97Wh at the 3.5-hour mark, which is 33.6% more than the baseline model with a remaining battery of 3.72Wh.

7.2.2 *Collaboration Environment*

We use the same sequence of contexts as in Section 7.2.1 for the collaboration environment. However, we modify the scenario so that the second confronting context has another camera presents with more battery remaining. In this scenario, the camera uses the weights and reward for more power savings. The results are shown in Figure 14. The controller chooses a lower resolution that saves some power but still records at a high frame rate. The remaining battery is 5.21Wh at 3.5-hour mark, which is 40.1% more than the baseline and 4.8% more than in Section 7.2.1.

8. CONCLUSION AND FUTURE WORK

In this paper, we propose a distributed adaptive model predictive controller for body cameras which improves information preservation while reducing energy consumption. The controller is adaptive and its performance improves over time. A fast and computation effective context recognition method is used in this system. We also presented a cost effective way to prototype a body camera which can be used in different research projects.

In the future, we intend to implement and fine tune the controller with more features on our body camera system and evaluate its performance in real world scenarios for a longer period of time. We plan to recruit more volunteers to collect IMU data and evaluate the effects of different parameters on the accuracy of classification.

9. REFERENCES

[1] Oresti Banos, Juan-Manuel Galvez, Miguel Damas, Alberto Guillen, Luis-Javier Herrera, Hector Pomares, Ignacio Rojas, Claudia Villalonga, Choong Seon Hong, and Sungyoung Lee. Multiwindow fusion for wearable activity recognition. In *International Work-Conference on Artificial Neural Networks*, pages 290–297. Springer, 2015.

[2] Ulf Blanke, Robert Rehner, and Bernt Schiele. South by south-east or sitting at the desk: Can orientation be a place? In *Wearable Computers (ISWC), 2011 15th Annual International Symposium on*, pages 43–46. IEEE, 2011.

[3] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33, 2014.

[4] Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Tröster. Robust recognition of reading activity in transit using wearable electrooculography. In *International Conference on Pervasive Computing*, pages 19–37. Springer, 2008.

[5] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[6] Gene F Franklin, J David Powell, Abbas Emami-Naeini, and J David Powell. *Feedback control of dynamic systems*, volume 3. Addison-Wesley Reading, MA, 1994.

[7] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practiceâĂŤa survey. *Automatica*, 25(3):335–348, 1989.

[8] David A Harris. Picture this: Body worn video devices ('head cams') as tools for ensuring fourth amendment compliance by police. 2010.

[9] Kai Kunze, Michael Barry, Ernst A Heinz, Paul Lukowicz, Dennis Majoe, and Jürg Gutknecht. Towards recognizing tai chi-an initial experiment using wearable sensors. In *Applied Wearable Computing (IFAWC), 2006 3rd International Forum on*, pages 1–6. VDE, 2006.

[10] Ioan Doré Landau, Rogelio Lozano, Mohammed M'Saad, and Alireza Karimi. *Adaptive control*, volume 51. Springer New York, 1998.

[11] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[12] Liuping Wang. *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, 2009.