2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems

A Context Aware Energy-Saving Scheme for Smart Camera Phones based on Activity Sensing

Yuanyuan Fan, Lei Xie, Yafeng Yin, Sanglu Lu
State Key Laboratory for Novel Software Technology, Nanjing University, P.R. China
Email: fyymonica@dislab.nju.edu.cn, lxie@nju.edu.cn, yyf@dislab.nju.edu.cn, sanglu@nju.edu.cn

***Abstract*—Nowadays more and more users tend to take photos with their smart phones. However, energy-saving continues to be a thorny problem for smart camera phones, since smart phone photographing is a very power hungry function. In this paper, we propose a context aware energy-saving scheme for smart camera phones, by accurately sensing the user's activities in the photographing process. Our solution is based on the observation that during the process of photographing, most of the energy are wasted in the preparations before the shooting. By leveraging the embedded sensors like the accelerometer and gyroscope, our solution is able to extract representative features to perceive the user's current activities including body movement, arm movement and wrist movement. Furthermore, by maintaining an activity state machine, our solution can accurately determine the user's current activity states and make the corresponding energy saving strategies. Experiment results show that, our solution is able to perceive the user's activities with an average accuracy of 95.5% and reduce the overall energy consumption by 46.5% for smart camera phones compared to that without energy-saving scheme.**

I. Introduction

Nowadays smart phones have been widely used in our daily lives. These devices are usually equipped with sensors such as the camera, accelerometer, and gyroscope. Due to the portability of smart phones, more and more people tend to take photos with their smart phones. However, energy-saving continues to be a upsetting problem for smart camera phones, since smart phone photographing is a very power hungry function. For example, according to KS Mobile's [1] report in 2014, the application *Camera 360 Ultimate* is listed in the first place of top 10 battery draining applications for Android. Therefore, the huge energy consumption becomes a non-negligible pain point for the users of smart camera phones.

Nevertheless, during the process of photographing, we observe that a fairly large proportion of the energy is wasted in the preparations before shooting. For example, the user first turns on the camera in the smart phone, then the user may move and adjust the camera phone time and again, so as to find a view, finally, the user focuses on the object and presses the button to shoot. A lot of energy is wasted in the process between two consecutive shootings, since the camera phone uses the same settings like the frame rate during the whole process, and these settings requires basically equal power consumption in the camera phone. Besides, it is also not wise to frequently turn on/off the camera function, since it is not only very annoying but also not energy efficient. Therefore, it is essential to reduce the unnecessary energy consumption during the photographing process to greatly extend the life time of smart camera phones. However, the previous work in energy-saving schemes for smart phones have the following limitations: First, they mainly reduce the energy consumption in a fairly isolated approach, without sufficiently considering the user's actual behaviors from the application perspective, this may greatly impact the user's experience of the smart phones. Second, in regard to the energy-saving scheme for photographing, they mainly focus on the shooting process instead of the preparations before shooting.

Corresponding Author: Dr. Lei Xie, lxie@nju.edu.cn

In this paper, we propose a context aware energy-saving scheme for smart camera phones, by accurately sensing the user's activities in the photographing process. Our idea is that, since current smart phones are mostly equipped with tiny sensors such as the accelerometer and gyroscope, we can leverage these tiny sensors to effectively perceive the user's activities, such that the corresponding energy-saving strategies can be applied according to the user's activities.

There are several challenges in building an activity sensing-based scheme for smart phones. The first challenge is to effectively classify the user's activities during the photographing process, which contains various levels of activities of bodies, arms and wrists. To address this challenge, we propose a three-tier architecture for activity sensing, including the body movement, arm movement and wrist movement. Furthermore, by maintaining an activity state machine, we can accurately determine the user's current activity states and make the corresponding energy saving strategies. The second challenge is to make an appropriate trade-off between the accuracy of activity sensing and energy consumption. In order to accurately perceive the user's activities with the embedded sensors, more types of sensor data and higher sampling rates are required. However, this further causes more energy consumption. To address this challenge, our solution only leverages those low power sensors, such as accelerometer and gyroscope, to classify the activities by extracting representative features to distinguish the user's activities respectively. We further choose sampling rates according to the user's current activities. In this way, we can sufficiently reduce the energy consumption of activity sensing so as to achieve the overall energy efficiency.

We make the following contributions in three folds: First, we propose a context aware energy-saving scheme for smart camera phones, by leveraging the embedded sensor to conduct activity sensing. Based on the activity sensing results, we can make the corresponding energy saving strategies. Second, we build a three-tier architecture for activity sensing, including the body movement, arm movement and wrist movement. We use low-power sensors like the accelerometer and gyroscope to extract representative features to distinguish the user's

978-1-4673-9101-6/15 $31.00 © 2015 IEEE
DOI 10.1109/MASS.2015.17

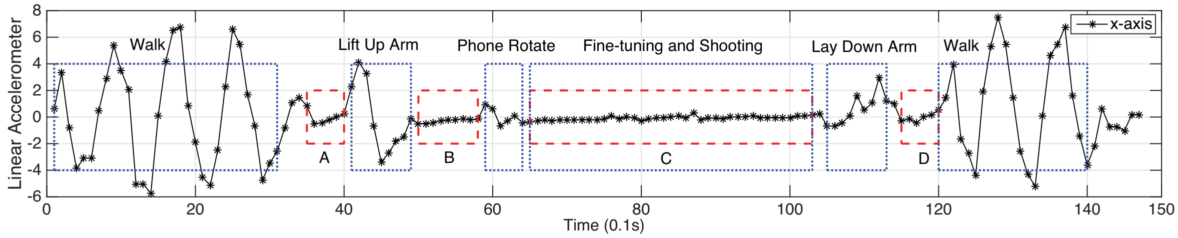


Fig. 1. The Process of Photographing

activities. By maintaining an activity state machine, we can classify the user's activities in a very accurate approach. Third, we have implemented a system prototype in android-powered smart camera phones, the experiment results shows that our solution is able to perceive the user's activities with an average accuracy of 95.5% and reduce the overall energy consumption by 46.5% for smart camera phones compared to that without energy-saving scheme.

II. System Overview

In order to adaptively reduce the power consumption based on activity sensing, we first use the built-in sensors of the phone to observe the human activities and discuss the energy consumption during the process of photographing. Then, we introduce the system architecture of our proposed energy-saving scheme.

A. Human activities related to photographing

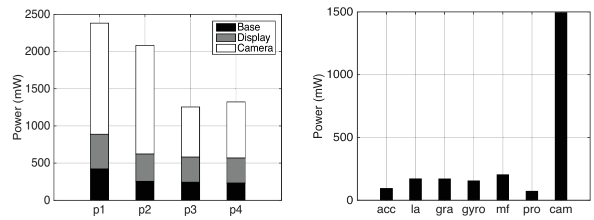
Usually, the users tend to have similar activities during photographing. As shown in Fig. 1, we use the linear accelerometer to detect the activities. Before or after the user takes photos, he/she may stay motionless, walking or jogging. While taking photos, the user usually lifts up the arm, rotates the phone, makes fine-tuning, shoots a picture, then lays down the arm. We categorize all the eight actions into the following three parts:

- **Body movement:** Motionless, walking and jogging
- **Arm movement:** Lifting up the arm and laying down the arm
- **Wrist movement:** Rotating the phone, making fine-tuning and shooting a picture

B. Energy consumption related to photographing

Before we propose the energy-saving scheme to reduce the power consumption based on the user's activities, we observe the energy consumption of the phone by using Monsoon power monitor [2] firstly.

1) Energy consumption in preparation for photographing: Power consumption for shooting a picture is large. We observe the power consumption in the following four android-based phones, i.e., Samsung GT-I9250, Huawei H30-T10, Samsung GT-I9230 and Huawei G520-T10. In Fig. 2(a), "Base" represents the phone's basic power when the phone is off. "Display" represents the power when phone is idle and only keeps screen on. "Camera" represents the power when camera works in



(a) Energy Consumption Components (b) Energy Consumption of Built-in Sensors

Fig. 2. Energy Consumption. (a) p1-Samsung GT-I9250, p2-Huawei H30-T10, p3-Samsung GT-I9023, p4-Huawei G520-T10; (b) acc-accelerometer, la-linear accelerometer, gra-gravity sensor, gyro-gyroscope, mf-magnetic field, pro-proximity, cam-camera.

preview mode. For the four phones, we can find that keeping screen on increases the power consumption by 20%, 18%, 27% and 26%, respectively. While using camera increases power consumption by 63%, 70%, 53% and 57%, respectively. Therefore, in preparation for photographing, energy consumption is wasted on camera and screen.

2) Energy consumption of turning on/off the camera: Frequently turning on/off the phone is annoying and energy wasting. When we need to take photos frequently for a period of time, the camera may switch between on/off frequently and so is to screen. And user needs to press button and unlock the screen with scrolling which results in large energy waste [3]. Based on Table I and Eq. (1), we can find that energy

TABLE I. Energy data of turning on/off phone (Samsung GT-I9250)

Subject / Energy	Turn On (Total)	Turn Off (Total)	Preview (1s)
Energy (uAh)	769.71	400.71	160.22

consumption of turning on/off the phone one time can keep camera working in previewing mode for seven seconds, which means one picture can be shot.

$$(769.71uAh + 400.71uAh)/160.22uAh = 7.305s \quad (1)$$

It indicates that *frequently turning on/off the phone manually is indeed inconvenient and rather energy-consuming.*

3) Energy consumption of the sensors: Fig. 2(b) shows the power consumption of a phone when a sensor is turned on. All these sensors work in their maximum sampling frequency, i.e., 100Hz. When we turn on the camera and phone will be in preview state, its increase of power is much larger than that of other sensors. Therefore, *low power-consuming sensors can be used to reduce the energy consumption of photographing.*

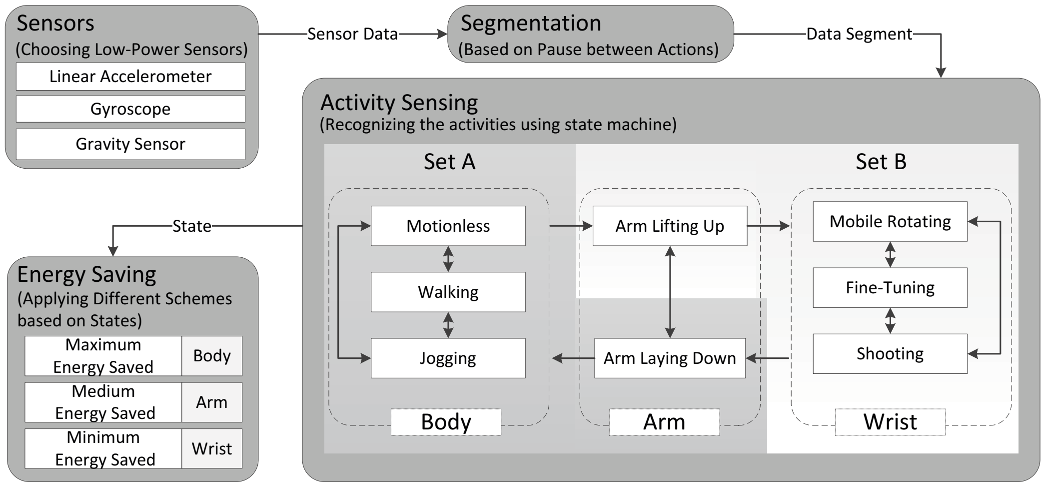


Fig. 3. System Architecture

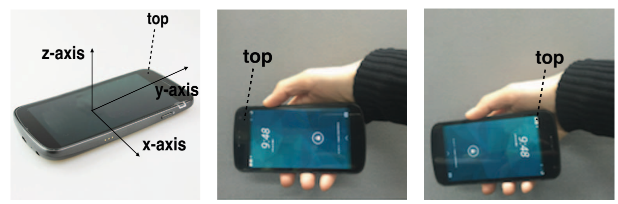
According to the above observations, we can utilize the low energy-consuming built-in sensors of the phone to detect the user's activities and reduce the energy consumption of taking photos. A simple example could be turning off the screen, decreasing the brightness of the screen, or decreasing the preview frame rate of the camera to reduce the energy cost when we find the user is not taking a photo.

C. System Architecture

The architecture of our system is shown in Figure 3. Firstly, we mainly obtain the data from low power-consuming built-in sensors, i.e., the linear accelerometer, the gyroscope and the gravity sensor, as shown in the "Sensor" module. Secondly, we separate the data into different regions, which are corresponding to the users' actions, as shown in the "Activity Sensing" module. Thirdly, we adaptively adopt an appropriate energy-saving scheme for each action, as shown in the "Energy Saving" module. In the following paragraphs, we briefly introduce how we can realize the activity sensing and reduce the power consumption.

1) Activity Sensing: Based on section II-A, the user's actions can be categorized into three parts: *body movement, arm movement, wrist movement*. Correspondingly, in our system architecture, we call the above parts as *body level, arm level, wrist level*, respectively. In each level, there may be more than one action. Besides, the different levels may exist some transfer relations. Therefore, we use the *State Machine* to describe the specific actions of the user. In the *State Machine*, each action is represented as a state. The transferable relations between the states are shown in Fig. 3. Before we determine the type of the action, we first estimate which level the action belongs to. Then, we further infer the specific action of the user based on more sensor information.

- **Body level:** It includes motionless, walking and jogging. Motionlessness can be recognized with its low variance of linear accelerometer's data. Then walking and jogging are distinguished with the frequency which can be calculated using Fast Fourier Transformation.



(a) Coordinates of Motion Sensors (b) hold horizontally naturally (c) hold horizontally backward

Fig. 4. Coordinates of Phone and Direction of Phone Hold

- **Arm level:** It includes lifting the arm up and laying the arm down. The relationship between the data of gravity sensor and linear accelerometer is used to distinguish the two actions. And voting mechanism is used to guarantee the accuracy.

- **Wrist level:** It includes rotating the phone, making fine-tuning, and shooting a picture. We make use of a linear SVM model to distinguish them with the variance, mean, maximum and minimum of three axises of three sensors as its features.

2) Energy-saving Scheme: Based on the feature and energy consumption in each action/state, we propose an adaptive energy-saving scheme for taking photos. For example, when you walk, jog or stay motionless, it's unnecessary to keep the screen on. When you lift your arm up, it's better to turn on the screen and adjust the screen's brightness based on the light conditions. When you make fine-tuning to observe the camera view before shooting a picture, it's better to make the camera work on the preview state. In this way, we can make the context aware energy-saving schemes for the camera phones.

III. System Design

In this section, we present the design of our energy-saving scheme for smart camera phones based on activity sensing.

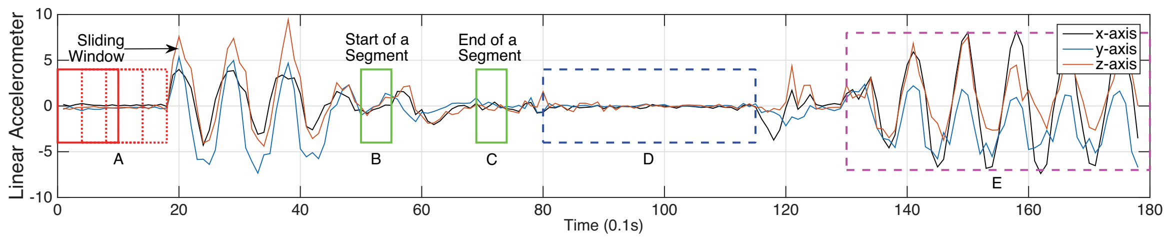


Fig. 5. Segment the Data of Linear Accelerometer

A. Activity Sensing

1) Raw Data Collection: We collect data from linear accelerometer, gyroscope and gravity sensor of android phones. These sensors have their own coordinates as shown in Figures 4(a), which are different from the earth coordinate system. For example, when we hold the phone as Fig. 4(b), the data of gravity sensor's x-axis almost equals to g ($9.8m/s^2$). When we hold the phone as Fig. 4(c) , the result is opposite.

2) Action Segmentation: From sensors, we can only get sequential raw data. To do the following activity sensing, data should be segmented as one segment corresponds to one action.

Observation. For a user, there is always a short pause between two different actions shown with red rectangles (A, B, and D) in Figure 1. However, some actions like fine-tuning and shooting are very gentle, it's difficult for the linear accelerometer to detect the pause from the actions shown with blue rectangle (D) in Figure 5. On this occasion, gyroscope is used to assist for the segmentation because it's more sensitive with the motion. The gyroscope data corresponding to the data in rectangle D is shown in Figures 6 and the pause between actions can be detected as shown with red rectangles (D1/D2). Back to Figure 5, one action which lasts for a long time shown with purple rectangle (E), may bring computational overhead.

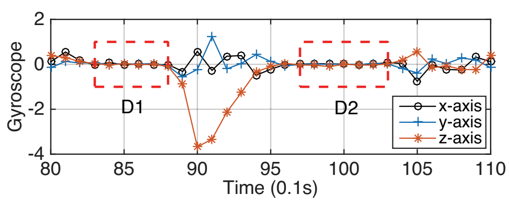


Fig. 6. Raw Data of Gyroscope Corresponding to Rectangle D in Fig 5

Segmentation. First, we leverage a *sliding window* to continuously calculate the *variances* of data of linear accelerometer's three axises. Second, if all three variances are below a threshold, the window is regraded as the start/end of a segment shown with green rectangle (B/C) in Fig 5. The window size is set as half of the value of sensor's sampling frequency as the pause between two continuous actions is always less than half a second. Third, when two continuous sliding windows whose variances are both below the threshold, we use the corresponding data of gyroscope and calculate the variance in the sliding window. If two continuous windows whose variances are below the threshold in gyroscope, we continue to calculate until a window whose variance above threshold is found. Then, this part is regarded as a segment. After that, we will take last sample of the window as a start point of the next segment and return back to calculate the variance in the window of linear accelerometer's data. Fourth, if there is too much data before the second eligible variance showing up, a maximum segment size is set to segment data. The maximum size is set as ten times of the value of sensor's sampling rate because most of the actions in arm and wrist level won't last for more than 10 seconds for common people.

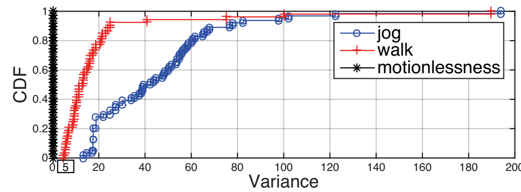


Fig. 7. CDF of Variance of Y-axis of Three Actions in Body Level

3) Action Recognition: We first do the recognition in three levels respectively. Then we describe how to do recognition among levels.

Body Level: Body level includes three actions which are motionlessness, walking and jogging. They are important actions which connect two shoots. As the movements of walking and jogging are very obvious, we take advantage of linear accelerometer to classify the actions.

Observation. Motionlessness is easy to be distinguished from walk and jog because of its *low variance* of raw data from linear accelerometer. Figures 7 shows the distribution of three actions' variances. Variance of motionlessness is almost zero and can be clearly distinguished. While walking and jogging can't be distinguished only based on the variances as they have some overlaps.

We hold the phone like Fig 4(b) and use linear accelerometer to get raw data of walking and jogging shown in Figures 8(a) and 8(c). We apply *Fast Fourier Transform* on the data and get the spectrum. Fig 8(b) shows that the frequency of walking is about 1 Hz. Fig 8(d) shows that the frequency of jogging is about 3.5 Hz. Thus, these two actions can be distinguished using frequency.

Recognition in Body Level. Effected by the holding gesture, the changes of data in three axises are different. (1) We first determine the axis whose data will be used. To common people, the phone can be held perpendicularly or parallel to the ground. If the phone is held perpendicularly to the ground, the data of z-axis doesn't change a lot in this level. If phone is held parallel to the ground, the data of x-axis doesn't change obviously. Therefore, we use the data of y-axis. (2) We

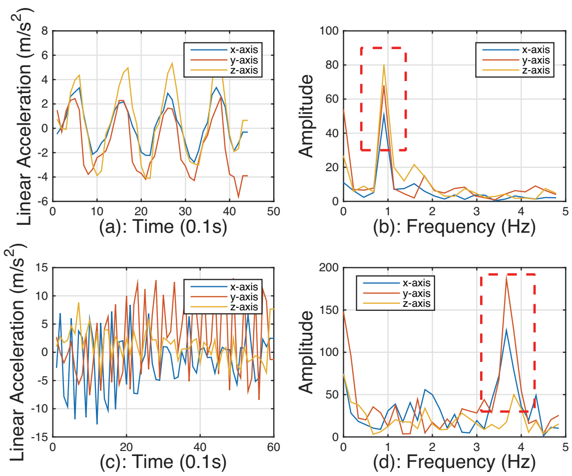


Fig. 8. Frequencies of walking and jogging. (a) shows raw data of walking and (b) shows its frequency. (c) shows raw data of jogging and (d) shows its frequency.

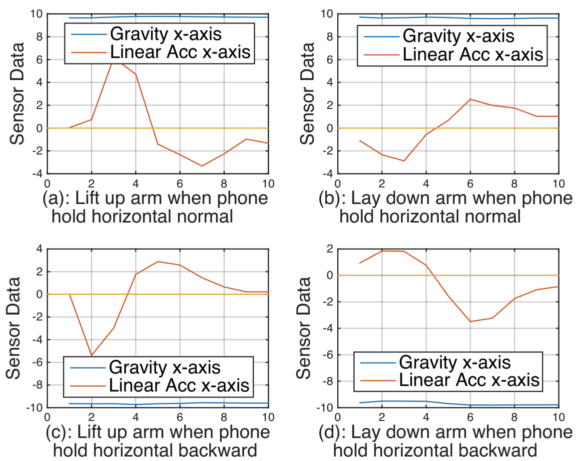


Fig. 9. Data of linear acceleration and gravity sensor of arm level when phone held horizontally in normal and in backward direction

calculate the variance of y-axis of linear accelerometer. If it is less than a threshold, the action is regarded as motionlessness. The threshold is set to 5 according to Figure 7. (3) If the action is not motionlessness, we apply FFT to the data segment. In general, the frequency of walk ranges from 1 Hz to 3 Hz and that of jog is 3 Hz to 6 Hz. Therefore, if the frequency is less than 3, the action is recognized as walking. Otherwise, it's jogging.

Arm Level: Arm level contains two actions, arm lifting up and laying down. They are actions which connect body and wrist level. After you lift up your arm, you may start shooting. After you lay down your arm, the shooting may end.

Observation. Arms lifting up and laying down are two reversed actions. When we hold the phone horizontally in normal direction as Fig 4(b), we get sensors' data of lifting arm up shown in Fig 9(a) and laying arm down shown in Figures 9 (b). Under this situation, the data of linear accelerometer will change mostly in x-axis as the motion happens in its

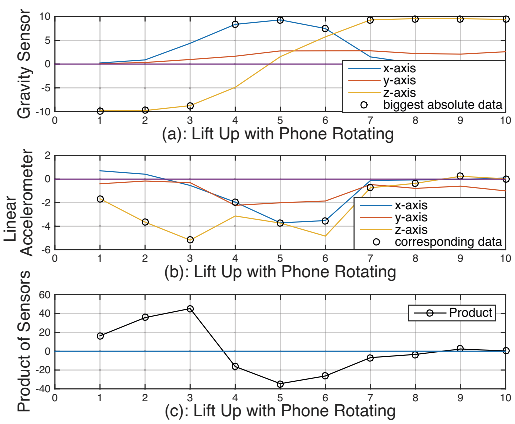


Fig. 10. Lift up the phone with rotating 360 degrees

direction. Therefore only the data of linear accelerometer's x-axis is showed. In Figures 9(a), when lifting up your arm, the value of gravity sensor stays positive while the value of linear accelerometer changes from positive to negative. It means the signs of two sensors' value change from same to different. In Figures 9(b), the signs of two sensors' value change from different to same. When the phone is held as Figures 4(c), the corresponding sensor data is showed in Figures 9(c) and (d). When lifting up your arm, the signs of x-axis value of two sensors still change form same to different. And when laying down your arm, the result is opposite.

However, the phone may be held in hand in any gesture. We lift up the phone and rotate 360 degrees at the same time. The data of gravity sensor and linear accelerometer is showed in Figures 10 (a) and (b). We can't simply figure out the relationship between two sensors. The data in three axises of gravity sensor, whose absolute value is maximum, is chosen as shown with black circle in Fig 10(a). And the data of linear accelerometer of corresponding axis is chosen, as shown with black circles in Fig 10(b). We multiply the two corresponding data and the result is showed in Figures 10(c). We can find that the sign changes from positive to negative. In summary, when you lift arm up, the signs of gravity sensor and linear accelerometer change from same to different. When you lay down arm, the change is diametrically opposite.

Recognition in Arm Level. The maximum absolute data of gravity sensor and the corresponding data of linear accelerometer are chosen. Then we analyze the relationship between the two sensors and then apply *voting mechanism* to avoid the noise made by hands tremble. At last, if the signs of two sensors' selected data change from same to different, the state should be arm lifting up. Otherwise, the state should be arm laying down. The specific process is showed in Algorithm 1.

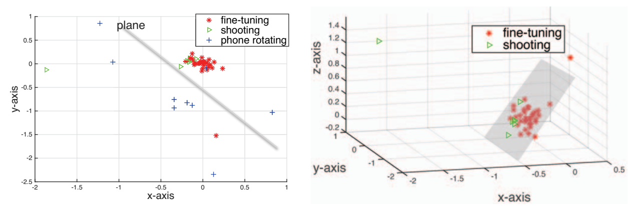
Wrist Level: Wrist movement contains phone rotating, fine-tuning and shooting. Picture is shot in this level.

Observation. The raw data of three axises of linear accelerometer of three actions are showed in Figures 11(a) and 11(b). From Figure 11(a), phone rotating can be distinguished by using a plane. From Figure 11(b), the other two actions can be distinguished. Therefore, a classifier as Support Vector Machine (SVM) can be used for classification. We take the

Algorithm 1: Recognition in Arm Level

Input: Data of three axises of gravity sensor and linear acceleromter
Output: Arm state

Calculate the absolute data of all three axises of gravity sensor
Get the axis position set A_i where absolute data is maximum among every three samples
Get the original gravity data set G_i according to axis position set A_i, get the linear accelerometer data set LA_i according to axis position set A_i
Multiply the corresponding data in set G_i and LA_i and store the sign of the result in set S_i
Voting the result in the first half of set S_i and store the sign result as $Sign_1$, voting the result in the second half and store the sign result as $Sign_2$
if *$Sign_1$ is positive and $Sign_2$ is negative* **then**
    return Lifting arm up
if *$Sign_1$ is negative and $Sign_2$ is positive* **then**
    return Laying arm down



(a) Data of Three Actions Shown in X-Y (b) Data of Fine-tuning and Shooting Shown in X-Y-Z

Fig. 11. Data of Linear Accelerometer of Three Actions in Wrist Level

characteristics of linear accelerometer, gyroscope and gravity sensor into account. Linear accelerometer can detect the absolute motion. Gyroscope is sensitive to the rotation of phone. Gravity sensor shows holding gesture. Therefore, using all these sensors' data to classify the actions can achieve good effect. The motions and movement range of these actions are different. Therefore, we extract the variance, mean, maximum, minimum of three axises as features and use linear kernel to train a SVM model to classify three actions. We also try to train the SVM model with other six combinations of sensors and three different kernels shown in Figure 12. And the accuracy of using three sensors with linear kernel is highest.

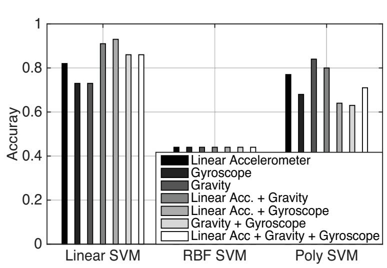


Fig. 12. Accuracies of Different SVM Models

Recognition in Wrist Level. We first calculate the variance, mean, maximum, minimum of all three axises of linear accelerometer, gyroscope and gravity sensor in the data segment then take advantage of trained SVM model to predict the current state.

Recognition among Three Levels: We use an *activity state machine* to help us do the activity recognition which is shown in Figure 3. All the eight actions are eight states and the arrows show the relationship among the actions. The eight states belong to three levels, which are body, arm and wrist. In each level, any two of the states can transform mutually.

However, the states in body level only can transform to arm lifting up state. And only arm laying down state can transform to the state in body level. The connection of arm and wrist level is similar. Based on the relationship, we divided all these states into two sets (set A and B) shown in Figure 3. What's more, once we press the button to take the picture, we calibrate the state if there's somthing wrong. Making use of the relationship and the calibration, we do activity recognition as shown in Algorithm 2.

Algorithm 2: Recognition among Three Levels

Input: Array A_i which stores eight states
Output: Current state C_s

Search A_i and get the last state l_s
Recognize in arm level and assign the result to Arm_s
if *Arm_s is in arm level* **then**
    C_s is assigned to Arm_s
else
    if *the button of shotting is pressed* **then**
        C_s is assigned to shooting
    else
        if *l_s is in Set A* **then**
            Recognize in body level and C_s is assigned to the result of recognition in body level
        else
            Recognize in wrist level and C_s is assigned to the result of recognition in wrist level
Update A_i
return C_s

B. Energy-saving Scheme

Based on the state obtained from activity sensing module and the feature of different action, an adaptive energy-saving scheme is proposed.

1) Energy Saving Points: Based on the analysis in section II-A, we have known the camera and screen are two energy-hungry parts.

Observation of Camera. To camera, the resolution of the photo and frame rate of preview are possible impact factors of energy consumption. However, high or low resolution has less effect to energy consumption shown in Fig. 13.

Before shooting, the camera is in preview mode. The frame rate of preview has relationship with energy consumption shown in Figures 14(a). We conduct the experiments with Samsung phone (GT-I9250). The x-labels represent the ranges of frame rate in preview mode. We discover that when the range is 15-15, the energy consumption is minimum. Because

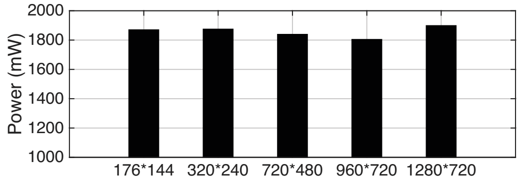
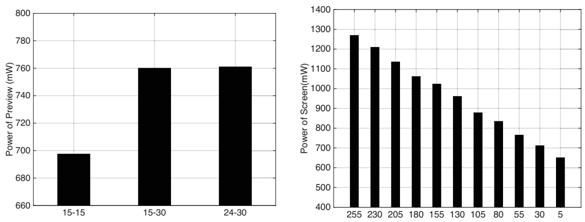


Fig. 13. Energy Consumption with Different Resolution of Sony LT26w



(a) Different Frame Rate of Preview (b) Different Brightness of Screen

Fig. 14. Energy Consumption of Different Brightness and Preview Frame Rates

of Android mechanism, the phone adaptively chooses the suitable preview frame rate in the range by itself. Therefore the energy consumption of last two situations are same.

Observation of Screen. Brightness of screen is related to the energy consumption shown in Figures 14(b). The range of brightness is 0-255. 0 stands for darkest and 255 for brightest. Once the brightness drops, the energy consumption decreases.

2) Energy Saving Scheme: For obtained states, we apply corresponding energy saving strategies, as shown in Figure 15.

If obtained state is in body level, the screen and the camera will be turned off as the user doesn't need to look at the screen. Further more, if the states always belong to body level for a long time (15 minutes is chosen in our implement), the sensors will be turned off until the camera software is used again.

If obtained state is in arm level, the screen is turned on and its brightness is adjusted based on ambient light as lifting up your arm. The brightness will be declined to 5 as laying down your arm. The brightness is set to five levels according to different environment shown in Table II.

TABLE II. THE BRIGHTNESS OF SCREEN IN DIFFERENT ENVIRONMENT

Number	Environment	Ambient Light (SI lux)	Brightness of Screen
1	Day, outdoor, sunny	>6000	180
2	Day, outdoor, cloudy	500∼6000	130
3	Day, indoor, no lamp	100∼300	80
4	Night, outdoor, street lamp	<100	55
5	Night, indoor, lamp	300∼500	105

If obtained state is in wrist level, the camera will be turned on and stay in preview mode. When you rotate the phone, camera is set to work with smallest frame rate supported by the phobe. In fine-tuning state, camera works with increased frame rate (median value is used). And in shooting state, all the indexes will return to normal. All the parameters can be changed by the user if the parameters do not fit them.

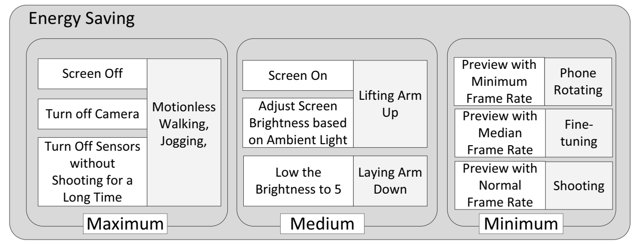


Fig. 15. Energy Saving Schemes Corresponding to Different States

IV. SYSTEM EVALUATION

We implement our system on Samsung GT-I9250 smartphone running on Google's Android platform. The version of the Android system is 4.4.2. We use Monsoon power monitor to measure the power consumption of the phone.

A. Impact of Sensors' Sampling Rate

1) Energy Consumption of Sensors with Various Sampling Rates: The sensors' maximum sampling rate of Samsung GT-I9250 is 100 Hz. We vary the sampling rate in twenty levels with step of 5. Energy consumption of sensors are different with various sampling rates as shown in Figure 16(a). We observe that power consumed is relatively big after 25 Hz.

2) Energy Consumption of Calculation of Sensors' Data: Energy consumption of calculation is related to the data size. With the sampling rate increasing, the energy also increases. We observe that the power of calculation of sliding window is only 1.5 mW. SVM model consumes about 4.34 mW as it is only used to do prediction with a trained model. The power of FFT is about 122 mW. The energy of other calculation can be ignored. Therefore, only energy of FFT is needed to be considered. But compared to the power of camera which is about 1500 mW, it is unobservable.

3) Activity Recognition Accuracies with Various Sampling Rates: We use six sampling rates, which are 2 samples/second, 5 samples/second, 10 samples/second, 20 samples/second, 50 samples/second and 100 samples/second, to evaluate the impact of sampling rate of sensors on the performance of recognition accuracy. In Figures 16(b), the accuracies of actions in body level are showed. For motionlessness, the accuracy has no relationship with the sampling rate. For walk, the accuracy is low with rate of 2 Hz and bigger than 90% with the other five rates. For jog, the accuracy is bigger than 90% at the rate of 10 Hz. In Figures 16(c), the accuracies of actions in arm level are showed. We can find out that the accuracy of 20 Hz is 100%. In Figures 16(d), the accuracies of actions in wrist level are showed and it is 100% when the sample rate is 100 Hz.

B. Trade off between Energy Consumption and Recognition Accuracy

From Figures 16(b)(c)(d), the accuracy can be 100% with 100 Hz. But it not energy efficiency as shown in Figure 16(a). Therefore, we need make a trade off between energy consumption and recognition accuracy.

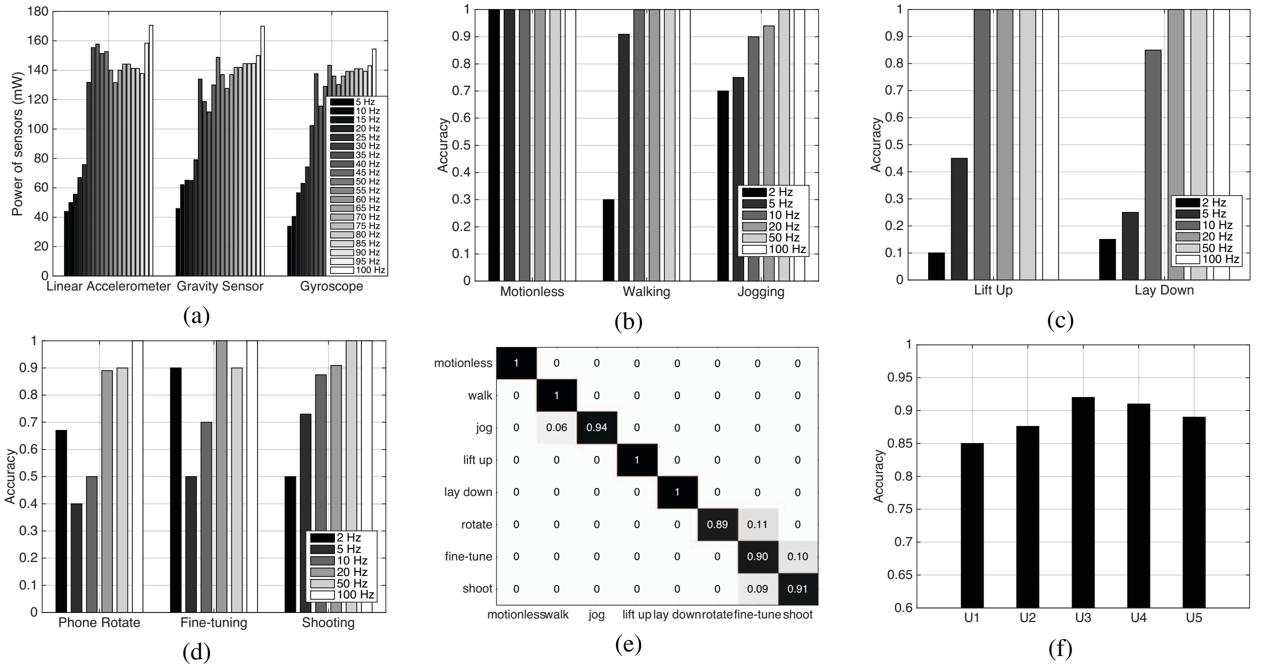


Fig. 16. Evaluation of Accuracy. (a) Sensors' sampling rates and corresponding energy consumption. (b) Sensors' sampling rates and accuracies of body level. (c) Sensors' sampling rates and accuracies of arm level. (d) Sensors' sampling rates and accuracies of wrist level. (e) Confusion matrix for eight actions. (f) Accuracies of different users.

In Figures 16(b), energy doesn't increase a lot when sampling rate changes from 10 Hz to 20 Hz while the accuracy of three actions can be increased to 94%. And it is similar to the actions in arm movement as the accuracy can be 100% at 20 Hz. In Figures 16(d), the accuracy is good when the sampling rate is 20 Hz. And the accuracy doesn't increase a lot if the sampling rate changes from 20 Hz to 50 Hz which will result in one times more energy consumption. Therefore, 20 Hz is used as sampling rate to recognize all the actions.

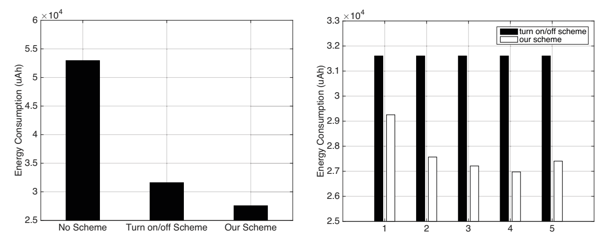
C. Recognition Accuracy

1) Recognition Accuracy of Our Scheme: The average accuracy of our scheme is 95.5%. Figure 16(e) plots the confusion matrix for eight actions with the sampling rate of 20 Hz. Each row denotes the actual actions performed by the user and each column the actions it was classified into. Each element in the matrix corresponds to the fraction of action in the row that were classified as the action in the column.

2) Recognition Accuracy of Different People: In order to evaluate the feasibility, we invite 5 users to test our smart camera in different environment. All the users use the phone to take 10 photos in five minutes. During the process, the users may perform any actions of three levels. The average accuracies of all ten processes are showed in Figures 16(f). We can find out that all the accuracies are above 85% and two of them are above 90%.

D. Energy Consumption

We measure the energy consumption under three schemes, which are no scheme, turn on/off scheme and our context-aware energy-saving scheme. We take 10 shoots in 5 minutes randomly in the same environment outdoor (cloudy) and the result is showed in Figures 17(a). Taking advantage of our scheme, energy consumption can be reduced 46.5% compared to no scheme and 12.7% compared to turn on/off scheme.



(a) 10 Shoots in 5 Minutes in Outdoor Using 3 Schemes
(b) 10 Shoots in 5 Minutes in 5 Different Environment

Fig. 17. Energy Consumption of the Process of Shooting using Different Schemes and in Different Environment

Then, we make measures when we use the smart camera phone in five different environment, as shown in Table II. In 17(b), the x-label respectively maps to the environment. Using our scheme, the energy consumption will change with the transformation of environment. Compared to turn on/off scheme, energy consumption decreases by 8%, 12.7%, 14%, 14.9% and 13.3% respectively.

V. Related Work

A. Energy Saving

Prior work on energy saving of smart-phone can be classified into three parts, analysis of hardwares' energy consumption [4], [5], [6], [7], power model [8][9] and energy saving schemes. Chen et al. [10] analyze the power consumption of AMOLED displays in multimedia applications. Camera recoding incurs high power cost. LiKamWa R et al. [11] do research on the image sensor and reveal two energy-proportional mechanisms which are supported by current image sensor but unused in mobile system. It indeed saves energy. But it only focuses on the energy consumption of the moment of shooting, while overlooking the consumption of preparation. Han et al. [12] study the energy cost made by human-screen interaction such as scrolling on screen. They propose a scrolling-speed-adaptive frame rate controlling system to save energy. Dietrich et al. [3] detect the game's current state and lower the processor's voltage and frequency whenever possible to save energy.

B. Activity Sensing

With the development of phone's built-in sensors, various approaches of activity recognition have been done. They can be classified into single-sensor and multi-sensors sensing.

Single sensor is used in the following work. Built-in microphone is used to detect the events that are closely related to sleep quality, including body movement, couch and snore [13]. Using built-in accelerometers, user's daily activities such as walking, jogging, upstairs, downstairs, sitting and standing are recognized in [14]. With the labeled accelerometer data, they apply four machine learning algorithms and make some analysis. Lee et al. [15] use accelerometers with hierarchical hidden markov models to distinguish the daily actions.

Multi-sensors are used in the following work. Shahzad et al. [16] propose a gesture based user authentication scheme for the secure unlocking of touch screen devices. It makes use of the coordinates of each touch point on the screen, accelerometer values and time stamps. Chen et al. [17] take advantage of features as light, phone situation, stationary and silence to monitor user's sleep. They need to use several different sensors to obtain all the phone's information. Driving style, which is concerned with man's life, is recognized by using the gyroscope, accelerometer, magnetometer, GPS and video[18]. Bo et al. [19] propose a framework to verify whether the current user is the legitimate owner of the smartphone based on the behavioral biometrics, including touch behaviors and walking pattens. These features are extracted from smartphone's built-in accelerometer and gyroscope.

VI. Conclusion and Future Work

In this paper, we propose a context aware energy-saving scheme for smart camera phone based on activity sensing. We take advantage of the features of activities and maintain an activity state machine to do the recognition. Then energy saving scheme is applied based on the result of recognition. Our solution can perceive the user's activities with an average accuracy of 95.5% and reduce the overall energy consumption by 46.5% for smart camera phones.

Following the current research, considering the difference between users, there are three possible directions for future work. First, more data of the process can be collected with our work to improve the design and implementation. Second, a self-constructive user preference learning can be designed to automatically extract the user perference of software settings. Third,to the phone whose configuration is too low, more simple strategy can be designed to avoid the possible delay for changing modes.

Acknowledgment

This work is supported in part by National Natural Science Foundation of China under Grant Nos. 61472185,61373129, 61321491, 91218302; Key Project of Jiangsu Research Program under Grant No. BE2013116; EU FP7 IRSES MobileCloud Project under Grant No. 612212. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization. Lei Xie is the corresponding author.

References

[1] "KS Mobile," http://www.cmcm.com/zh-cn/.

[2] "Monsoon PowerMonitor," https://www.msoon.com/LabEquipment/PowerMonitor/.

[3] B. Dietrich and S. Chakraborty, "Power management using game state detection on android smartphones," in *Proc. of ACM MobiSys*, 2013.

[4] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proc. of ACM SIGARCH*, 2007.

[5] F. Bellosa, A. Weissel, M. Waitz, and S. Kellner, "Event-driven energy accounting for dynamic thermal management," in *Proc. of COLP*, 2003.

[6] D. Rajan, R. Zuck, and C. Poellabauer, "Workload-aware dual-speed dynamic voltage scaling," in *Proc. of IEEE RTCSA*, 2006.

[7] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proc. of ACM SIGCOMM*, 2009.

[8] M. Dong and L. Zhong, "Self-constructive high-rate system energy modeling for battery-powered mobile systems," in *Proc. of ACM MobiSys*, 2011.

[9] F. Xu, Y. Liu, Q. Li, and Y. Zhang, "V-edge: Fast self-constructive power modeling of smartphones based on battery voltage dynamics." in *Proc. of NSDI*, 2013.

[10] X. Chen, Y. Chen, Z. Ma, and F. C. Fernandes, "How is energy consumed in smartphone display applications?" in *Proc. of ACM HotMobile*, 2013.

[11] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl, "Energy characterization and optimization of image sensing toward continuous mobile vision," in *Proc. of ACM MobiSys*, 2013.

[12] H. Han, J. Yu, H. Zhu, Y. Chen, J. Yang, G. Xue, Y. Zhu, and M. Li, "E 3: energy-efficient engine for frame rate adaptation on smartphones," in *Proc. of ACM Sensys*, 2013.

[13] T. Hao, G. Xing, and G. Zhou, "isleep: unobtrusive sleep quality monitoring using smartphones," in *Proc. of ACM Sensys*, 2013.

[14] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD*, vol. 12, no. 2, pp. 74–82, 2011.

[15] Y.-S. Lee and S.-B. Cho, "Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer," in *Proc. of Springer HAIS*, 2011.

[16] M. Shahzad, A. X. Liu, and A. Samuel, "Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it," in *Proc.of ACM MobiCom*, 2013.

[17] Z. Chen, M. Lin, F. Chen, N. D. Lane, G. Cardone, R. Wang, T. Li, Y. Chen, T. Choudhury, and A. T. Campbell, "Unobtrusive sleep monitoring using smartphones," in *Proc. of IEEE PervasiveHealth*, 2013.

[18] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proc. of IEEE ITSC*, 2011.

[19] C. Bo, L. Zhang, X.-Y. Li, Q. Huang, and Y. Wang, "Silentsense: silent user identification via touch and movement behavioral biometrics," in *Proc. of ACM MobiCom*, 2013.