Motion Sensors for Activity Recognition in an Ambient-Intelligence Scenario

Pietro Cottone, Giuseppe Lo Re, Gabriele Maida and Marco Morana

DICGIM - University of Palermo
Viale delle Scienze, ed. 6 - 90128 Palermo, Italy
{*pietro.cottone, giuseppe.lore, gabriele.maida, marco.morana*}*@unipa.it*

***Abstract*—In recent years, Ambient Intelligence (AmI) has attracted a number of researchers due to the widespread diffusion of unobtrusive sensing devices. The availability of such a great amount of acquired data has driven the interest of the scientific community in producing novel methods for combining raw measurements in order to understand what is happening in the monitored scenario. Moreover, due the primary role of the end user, an additional requirement of any AmI system is to maintain a high level of pervasiveness. In this paper we propose a method for recognizing human activities by means of a time of flight (ToF) depth and RGB camera device, namely Microsoft Kinect. The proposed approach is based on the estimation of some relevant joints of the human body by using Kinect depth information. The most significative configurations of joints positions are combined by a clustering approach and classified by means of a multi-class Support Vector Machine. Then, Hidden Markov Models (HMMs) are applied to model each activity as a sequence of known postures. The proposed solution has been tested on a public dataset while considering four different configurations corresponding to some state-of-the-art approaches and results are very promising. Moreover, in order to maintain a high level of pervasiveness, we implemented a real prototype by connecting Kinect sensor to a miniature computer capable of real-time processing.**

***Keywords*-Activity Recognition, Ambient Intelligence;**

I. Introduction

In recent years, the availability of an ever-increasing number of cheap and unobtrusive sensing devices, has piqued the interest of the scientific community in producing novel methods for understanding what is happening in the environment according to the acquired raw measures.

In particular, Ambient Intelligence (AmI) is a new paradigm in Artificial Intelligence that aims at exploiting the information about the environment state in order to personalize it, that is to adapt the environment to the user preferences [1].

The personalization process should be invisible to the user, thus the intrinsic requirement of any AmI system is the presence of pervasive sensory devices.

In our architecture, the sensory component is implemented through a Wireless Sensor and Actuator Network (WSAN), whose nodes are equipped with off-the-shelf sensors (i.e., outdoor temperature, relative humidity, ambient light exposure and noise level). Such networks extend the monitoring functionalities provided by traditional WSNs [2] since they include an active part, i.e., the actuators, that allows to modify the environment according to the observed data, high-level goals (e.g., energy efficiency) and user preferences [3]. Even if traditional sensors allow to understand the environment characteristics the user prefers, in order to perceive high-level features such as *what* the user is doing, the most functional devices are video sensors.

In this work we present a system for the management of an office environment, namely the rooms of a university department, using a time of flight depth and RGB camera device, i.e. Microsoft Kinect, to unobtrusively perform activity recognition. We started from the OpenNI / NITE APIs [4], [5] which provide an efficient global skeleton detection method that allows to represent a human body as a set of connected joints. Thus, a specific body posture can be considered as a particular configuration of connected joints and human activities can be described as spatio-temporal evolutions of different body postures.

In the scenario we are considering, unobtrusive sensor nodes are deployed in various rooms close to *sensitive* indoor areas. In order to preserve the pervasiveness of the system, the motion detection sensor provided by Kinect is coherently connected to a miniature fanless computer with reduced computation capabilities.

The paper is organized as follows: some related works are presented in Section II, while the proposed system architecture is described in Section III. The experimental scenario will be discussed in Section IV. Conclusions will follow in Section V.

II. Related Work

During the past few years, the issue of human action recognition has been addressed in several works.

In [6], the authors use a set of binary silhouettes as input of a framework based on Hidden Markov Models. An activity is described as a sequence of the poses of the person. The silhouettes are extracted from video images, thus this method lacks of flexibility since it requires a number of image processing steps (e.g., background removal, vector quantization, image normalization).

Two different recognition systems based on Silhouette features and Discrete Hidden Markov Models are presented in [7], [8]. The authors of [7] use Fourier shape descriptors,

978-1-4244-9529-0/13/$31.00 ©2013 IEEE

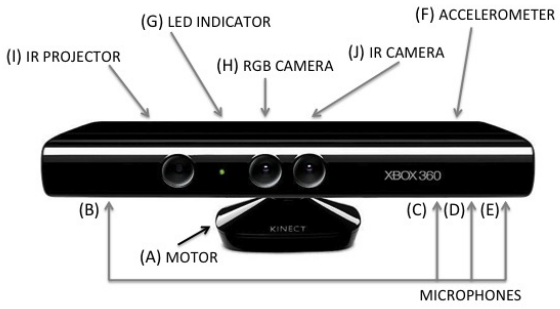

Figure 1: Kinect components.

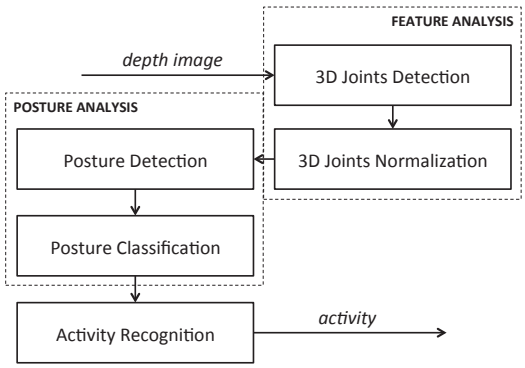

Figure 2: System Overview.

while in [8] the features are obtained by combining RGB and depth information. In both works, feature classification is performed by Support Vector Machines and the classified postures are considered as the discrete symbols emitted from the hidden states.

Several works [9], [10] address the problem of activity recognition by using intrusive sensors, e.g., wearable sensors. The release of the Kinect sensor allowed researchers to perform activity recognition in a unobtrusive way, i.e., by using depth and RGB information.

In [11], salient postures are characterized as a bag of 3D points obtained from the depth map. Such postures represent the nodes in an activity graph that is used to model the dynamics of the activities.

A model for human actions called Actionlet Ensemble Model is presented in [12]. Human bodies are considered as a large number of kinematic joints and actions are characterized by the interaction of a subset of these joints. The authors introduced the concept of Actionlet as a particular conjunction of the features for a subset of joints. As there is an enormous number of possible Actionlets, a data mining approach is applied to discover the discriminative Actionlets. Then an action is represented as an Actionlet Ensemble, which is a linear combination of the Actionlets.

A supervised algorithm that use a dictionary of labelled hand gestures is presented in [13]. The authors use Kinect SDK to extract a sequence of skeleton-model parameters that represents the feature space. The covariance matrix of this space is used to discriminate the gestures and action recognition is performed by a NN classifier.

A histogram based representation of human postures is presented in [14]. In this representation, the 3D space is partitioned into n bins using a spherical coordinate system. The authors built a model of human postures on 12 selected joints. Each joint position belongs to a bin with a certain level of uncertainty. The set of the vectors from the training sequences are reprojected using LDA and clustered into a K-postures vocabulary. The activities are represented as sequences of postures in the vocabulary and are recognized using HMM classifiers.

III. System Overview

According to the considered scenario, we found that Kinect represents the most suitable device both in terms of cost and functionalities since it is equipped with ten input/output components (see Fig. 1) that make it possible to sense the users and their interaction with the surrounding environment. The Kinect sensor rests upon a base which contains a motor (Fig. 1-**A**) that allows to control the tilt angle of the cameras (30 degrees up or down). Starting from the bottom of the device, you can see three adjacent microphones on the right side (Fig. 1-**C-D-E**), while a fourth microphone is placed on the left side (Fig. 1-**B**). A 3- axis accelerometer (Fig. 1-**F**) can be used to measure the position of the sensor, while a led indicator (Fig. 1-**G**) shows its state. However, the core of the Kinect is represented by the vision system composed of: an RGB camera (Fig. 1-**H**) with VGA standard resolution (i.e., 640x480 pixels); an IR (Fig. 1-**I**) projector that shines a grid of infrared dots over the scene; an IR (Fig. 1-**J**) camera that captures the infrared light. The factory calibration of the Kinect makes it possible to know the exact position of each projected dot against a surface at a known distance from the camera. Such information is then used to create depth images of the observed scene (i.e., pixel values represent distances) that capture the object position in a three-dimensional space.

The proposed system (see Fig. 2) aims at automatically inferring the activity performed by the user according to a set of known postures. Each posture is defined by the position of some body joints extracted by means of the OpenNI/NITE skeleton detection method. The set of detected joints is clustered by applying the K-Means algorithm in order to build a vocabulary of postures. The obtained "words" are validated by Support Vector Machines (SVMs). Finally, Hidden Markov Models (HMMs) are applied to model each activity as a sequence of vocabulary words.

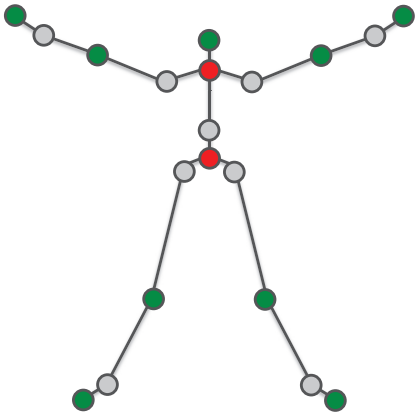

Figure 3: The 20 joints of the human body. Reference joints (red): *neck*, *hip center*. Selected joints (green): *head*, *elbows*, *hands*, *knees*, *feet*. Discarded joints (grey): *shoulders*, *wrists*, *spine*, *hips*, *ankles*.

A. Features Analysis

The OpenNI/NITE skeleton detection method is able to perform real-time detection (i.e., to find the 3D coordinates) of 20 body joints (see Fig. 3). However, due to the sensitiveness of the IR sensor, some overlaying detected joints (e.g., hands touching other body parts) or occlusions (e.g., objects placed between the sensor and the user) may lead to significant errors.

For this reason, some redundant joints (i.e., wrists, ankles) have been discarded due to their closeness to other selected joints (i.e., hands, feet), while others (i.e., spine, neck, hip and shoulders) are not relevant for the activity recognition. The selected joints are shown in green in Fig. 3, while the discarded ones in grey.

Moreover, since the distance of the skeleton joints from the hip depends on several things (e.g., the users height, arm length, distance from the sensor), all feature vectors have been normalized according to the distance between the neck and hip center joints. A scale-independent representation of the body posture is then obtained by fixing the center of the reference coordinate system at the hip center and considering as x-direction the left-right hip axis. Reference joints are shown in red in Fig. 3.

B. Postures Analysis

Once the joints have been detected, a clustering algorithm is applied to quantize the number of observed joints configurations. Thus, the detected features are clustered into K classes (i.e., building a K-words vocabulary) by using the K-means algorithm. Each posture is then represented as a single word of the vocabulary and therefore each activity can be considered as an ordered sequence of vocabulary words.

In order to obtain a better statistical description of the content of each cluster, the output (i.e., the pairs features/cluster) of the K-means algorithm is used to train a multi-class SVM. SVMs are supervised learning models used for binary classification and regression. A multi-class SVM is a net of SVMs able to perform a multi-class classification [15].

Moreover, since we transform sequences of joints configurations into the corresponding sequence of K-words, we consider only postures transitions, that is all repeated sequences of the same posture are merged. Thus, we obtain a more compact representation of the sequences addressing the problem of recognizing the same activities performed with different time durations.

C. Activity Recognition

In order to address the issue of recognizing different sequences of postures referred to the same activity, a probabilistic approach has been applied. In particular, we modelled each action using a discrete Hidden Markov Model (HMM) [16].

A HMM that has N states $S = \{S_1, S_2, ..., S_N\}$ and M output symbols $V = \{v_1, v_2, ..., v_M\}$ is fully specified by the triplet $\lambda = \{A, B, \pi\}$. The state transition probability distribution $A = \{a_{i,j}\}$ is

$$a_{i,j} = P\left[q_{t+1} = S_j | q_t = S_i\right], \quad 1 \leq i, j \leq N \tag{1}$$

where q_t is the actual state at time t.

The observation symbol probability distribution in state j, $B = \{b_j(k)\}$ is

$$b_j(k) = P\left[v_k \ \texttt{at} \ t | q_t = Sj\right], \tag{2}$$

where $1 \leq j \leq N$ and $1 \leq k \leq M$.

And the initial state distribution $\pi = \{\pi_i\}$ is

$$\pi_i = P\left[q_1 = S_i\right], \quad 1 \leq i \leq N \tag{3}$$

Once each HMM has been trained on the posture sequences of each activity, a new (unknown) sequence is tested against the set of HMMs and classified according to the largest posterior probability, if such a probability rises above a prefixed threshold.

IV. Results

The proposed method is part of a system aiming for timely and ubiquitous observations of an office environment, namely a department building, in order to fulfil constraints deriving both from the specific user preferences and from considerations on the overall energy consumption.

The system will reason on high-level concepts as “air quality”, “lighting conditions”, “room occupancy level”, each one referring to a physical measurement captured by a physical layer. Since the system must be able to learn the

Activity Set 1	Activity Set 2	Activity Set 3
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward Kick
Forward punch	Draw x	Jogging
High throw	Draw tick	Tennis swing
Hand Clap	Draw Circle	Tennis serve
Tennis serve	Two hand wave	Golf swing
Pickup & throw	Side boxing	Pickup & throw

Table I: The three Activity Sets.

user preferences, ad-hoc sensors for capturing the interaction between users and actuators are needed similarly to what is described in [17].

We evaluated the proposed activity recognition approach on the public MSR Action3D dataset [11] containing 20 actions: *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing and pickup & throw*. Every action is repeated 3 times by 10 different subjects.

During the training phase, we noticed that the skeleton tracker heavily failed in correspondence of some particular actions or subjects, as reported by the authors of the dataset[1]. For this reason, the *"bend"* and *"side kick"* actions and the subject 4 have been removed. Thus our filtered dataset is made up by 18 actions performed by 9 subjects.

Three Activity Sets (ASs) have been obtained from the filtered dataset in similar way as done by [11] and [14]. Each Activity Set contains 7 activities as shown in Table I.

Since, a number of solutions based on the SVM-HMM chain are presented in literature, we decided to verify how each processing module affects the overall system performance. For this reason, the proposed methods has been tested against four different system configurations.

In the first, *NONE configuration*, the posture analysis is performed by applying only the K-means algorithm; in the second, *PCA configuration*, a PCA transformation on original data (i.e., joints positions) has been added to the feature analysis process in order to evaluate the impact of a reduced feature space on the system performance; in the third, *SVM configuration*, we performed posture classification by means of a multi-class SVM classifier based on a RBF kernel with $\gamma = 1/Num.features$, and regularization parameter $C = 1$; in the last, *SVM_PCA configuration*, the impact of using both PCA and SVM has been evaluated.

The number of posture clusters (K) and HMM states (N) were founded by a Grid Search [18] in the range $[10; 100]$ for K and $[3; 8]$ for N. For every node of the grid, the error of Leave One Out Cross Validation [19] was computed. For each of the three Activity Set, 188 action sequences were

[1]http://research.microsoft.com/%7Ezliu/ActionRecoRsrc

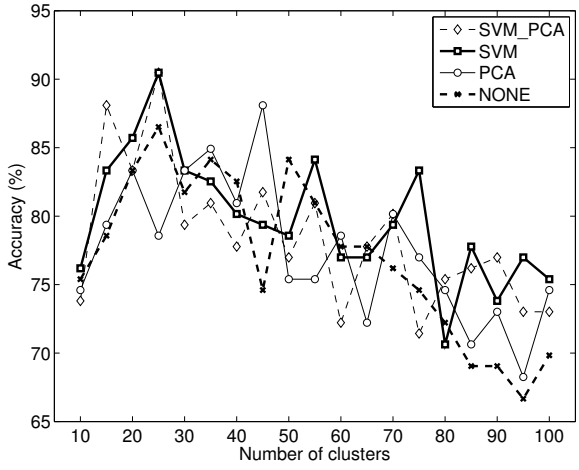

Figure 4: Comparison of the mean accuracy computed on the three Activity Sets while using four system configurations.

Configuration	(K,N)	Accuracy
NONE	(25,4)	86.50%
PCA	(45,7)	88.09%
SVM	(25,5)	90.47%
PCA_SVM	(25,10)	90.47%

Table II: Best mean accuracy obtained for each configuration.

used for training and the remaining sequence was used for validation; each test for a particular configuration set was repeated 10 times. As a result of these experiments we have chosen the pair (K, N) that minimizes the mean error on the three Activity Sets.

The results obtained for the best (K, N) pairs of each configuration are reported in Table II. The reduction of the feature space, obtained by applying Principal Component Analysis on original data, decreased the system performances. This result is motivated by the joints selection we preliminary performed, demonstrating that the original feature space does not contain correlated features.

A comparison of the accuracy measured while varying the number of clusters is shown in Fig. 4. The best performances are obtained by *SVM* and *SVM_PCA*, both giving an overall mean accuracy of 90.47%. However, the results obtained by the *SVM configuration* showed a smaller variance, demonstrating that the former is preferable.

Such a result is confirmed by comparing *SVM* and *SVM_PCA* on different values of K, as showed in Fig. 5. Moreover, according to the Minimum Description Length (MDL) [20], the model given by *SVM* is better than the *SVM_PCA* one, since the former uses a smaller number of states (i.e., $N = 5$ versus $N = 10$) as shown in Table II.

In table III are reported the mean accuracy values obtained

Action	Accuracy	Action	Accuracy
Horizontal arm wave	100%	Hand catch	71%
Hammer	100%	Two hand wave	100%
Forward punch	100%	Draw x	68%
Golf swing	83%	Draw tick	100%
Hand Clap	95%	Draw Circle	68%
Tennis serve	92%	High arm wave	83%
Pickup & throw	100%	Side boxing	83%
High throw	84%	Forward Kick	100%
Jogging	100%	Tennis swing	100%
Mean Accuracy 90.4%			

Table III: Recognition rate of SVM system configuration

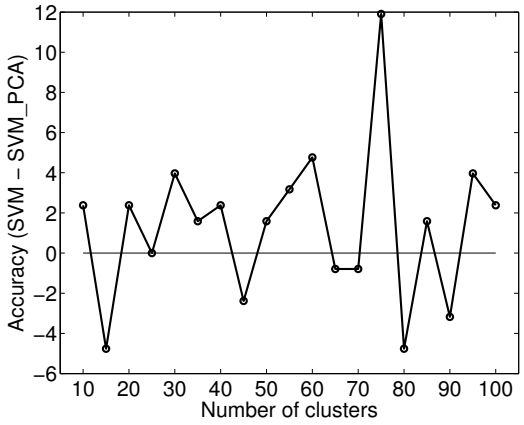

Figure 5: Difference of accuracy between the proposed system configuration *SVM* and *SVM_PCA*.

by the *SVM configuration* for the whole set of considered activities.

The confusion matrices reported in Table IV - V - VI show classification errors related to the three activity datasets listed in Table I. Please note that some activities are not correctly classified since they are considered as parts of more complex ones (e.g., *Hand catch* gesture is the beginning of *High arm wave*, *Draw tick* and *Two hand wave*).

The overall system has been tested using MATLAB and LIBSVM [21]. A prototype of the activity recognition module has been implemented connecting the Kinect to a miniature fanless PC (i.e., a fit-PC2i with Intel Atom Z530 1.6GHz CPU and Linux OS with kernel 2.6.32) that guarantees real-time processing of the observed scene with minimum levels of obtrusiveness and low power consumptions. The MATLAB implementation of our system takes a mean recognition time (i.e., posture analysis and activity recognition) of 1.25 seconds, with a power consumption of about 8W, while during idle the consumption is about 6W.

V. Conclusion

In this work we presented a system for the management of an office environment, namely the rooms of a university department, using Microsoft Kinect to unobtrusively perform activity recognition. We considered a scenario where the whole environment is permeated with small pervasive sensor devices, for this reason the Kinect is coherently connected to a miniature fanless computer with reduced computation capabilities. The proposed system infers the activity performed by the user according to a set of known postures, automatically extracted from training data by using a K-Means approach and SVM classification. Each activity is modelled by a Hidden Markov Models (HMMs) built on postures sequences.

	1	2	3	4	5	6	7
1	100	-	-	-	-	-	-
2	-	100	-	-	5	10	-
3	-	-	100	-	-	-	-
4	-	-	-	84	-	-	-
5	-	-	-	16	95	-	-
6	-	-	-	-	-	90	-
7	-	-	-	-	-	-	100
8	-	-	-	-	-	-	-

Table IV: Confusion matrix of Activity Set 1. (1) Horizontal arm wave, (2) Hammer, (3) Forward punch, (4) High throw, (5) Hand Clap, (6) Tennis serve, (7) Pickup & throw, (8) Unknown.

	1	2	3	4	5	6	7
1	83	9	-	-	-	-	-
2	-	71	-	-	-	-	-
3	-	-	68	-	-	-	7
4	-	11	-	100	12	-	-
5	-	-	32	-	68	-	-
6	17	9	-	-	10	100	-
7	-	-	-	-	-	-	83
8	-	-	-	-	10	-	10

Table V: Confusion matrix of Activity Set 2. (1) High arm wave (2) Hand catch, (3) Draw x, (4) Draw tick, (5) Draw Circle, (6) Two hand wave, (7) Side boxing, (8) Unknown.

	1	2	3	4	5	6	7
1	84	-	-	-	-	-	-
2	-	100	-	-	-	-	-
3	-	-	100	-	-	-	-
4	-	-	-	100	-	-	-
5	-	-	-	-	94	-	-
6	-	-	-	-	-	83	-
7	16	-	-	-	-	-	100
8	-	-	-	-	6	17	-

Table VI: Confusion matrix of Activity Set 3. (1) High throw (2) Forward Kick , (3) Jogging, (4) Tennis swing, (5) Tennis serve, (6) Golf swing, (7) Pickup & throw, (8) Unknown.

The activity models are independent of who performs the actions, independent of the speed at which the actions are performed, scalable to a large number of actions, and expandable with new actions. Moreover, since all repeated sequences of the same posture are merged, the proposed method is able to recognize the same activities performed with different time durations.

We validated the proposed solution against four different system configurations demonstrating that the use of SVM significantly improves the accuracy. We chose MSR Action3D dataset since it is hard to find reliable activity dataset addressing the same scenario we considered. However, the results we obtained and the particular architecture of our system make clear that the proposed approach can be easily adapted to recognize more complex activities by decomposing them in a chain of simpler ones.

We are actually working on the construction of a real prototype of the monitoring and controlling system allowing for exhaustive testing. Finally, we noticed that in many cases the quality of existing public dataset is poor, once that the collected data will be analysed and verified, we intend to publish both our dataset and results.

ACKNOWLEDGMENT

This work is supported by the SMARTBUILDINGS project, funded by POR FESR SICILIA 2007-2013.

REFERENCES

[1] A. De Paola, M. La Cascia, G. Lo Re, M. Morana, and M. Ortolani, "User Detection through Multi-Sensor Fusion in an AmI Scenario," in *Proc. of the 15th International Conference on Information Fusion*. Published by the IEEE Computer Society, 2012, pp. 2502–2509.

[2] G. Anastasi, G. Lo Re, and M. Ortolani, "Wsns for structural health monitoring of historical buildings," in *Human System Interactions, 2009. HSI '09. 2nd Conference on*, may 2009, pp. 574 –579.

[3] A. De Paola, S. Gaglio, G. Lo Re, and M. Ortolani, "Sensor9k: A testbed for designing and experimenting with WSN-based ambient intelligence applications," *Pervasive and Mobile Computing. Elsevier*, vol. 8, no. 3, pp. 448–466, 2012.

[4] PrimeSense, "Openni," http://www.primesense.com/en/openni.

[5] ——, "Nite," http://www.primesense.com/Nite.

[6] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, jun 1992, pp. 379 –385.

[7] M. P. V. Kellokumpu and J. Heikkila, "Human activity recognition using sequences of postures," in *In Proc IAPR Conf. Machine Vision Applications*, 2005, pp. 570 – 573.

[8] M. Tang, "Recognizing hand gestures with microsoft's kinect," , Department of Electrical Engineering, Stanford University, Tech. Rep., march 2011.

[9] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensorsa review of classification techniques," *Physiological Measurement*, vol. 30, no. 4, p. R1, 2009.

[10] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, A. Ferscha and F. Mattern, Eds. Springer Berlin Heidelberg, 2004, vol. 3001, pp. 1–17.

[11] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, june 2010, pp. 9 –14.

[12] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, june 2012, pp. 1290 –1297.

[13] K. Lai, J. Konrad, and P. Ishwar, "A gesture-driven computer interface using kinect," in *Image Analysis and Interpretation (SSIAI), 2012 IEEE Southwest Symposium on*, april 2012, pp. 185 –188.

[14] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, june 2012, pp. 20 – 27.

[15] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[16] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 –286, feb 1989.

[17] A. De Paola, G. Lo Re, M. Morana, and M. Ortolani, "An Intelligent System for Energy Efficiency in a Complex of Buildings," in *Proc. of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability*, 2012.

[18] S. S. Rao, *Engineering Optimization: Theory and Practice, 3rd Edition*. Wiley-Interscience, 1996.

[19] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, no. 0, pp. 40–79, 2010.

[20] P. D. Grünwald, *The Minimum Description Length Principle (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.

[21] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/%7Ecjlin/libsvm.