Towards A Unified System for Multimodal Activity Spotting: Challenges and A Proposal

Long-Van Nguyen-Dinh
Wearable Computing Lab
ETH Zürich
Zürich, Switzerland
longvan@ife.ee.ethz.ch

Gerhard Tröster
Wearable Computing Lab
ETH Zürich
Zürich, Switzerland
troester@ife.ee.ethz.ch

Alberto Calatroni
Wearable Computing Lab
ETH Zürich
Zürich, Switzerland
alberto.calatroni@ife.ee.ethz.ch

UbiComp '14, September 13 - 17 2014, Seattle, WA, USA

ACM 978-1-4503-3047-3/14/09$15.00.
http://dx.doi.org/10.1145/2638728.2641301

Abstract

In the existing multimodal systems for activity recognition, there is no single method to process different sensor modalities at different on-body positions. Moreover, sensor types are often selected and optimized so as to accord with the goal of application. The complexity makes those systems infeasible to be deployed for new settings. This paper proposes a unified system which works with any available wearable sensors placed on user's body to spot activities. Each data stream is treated uniformly through our proposed template matching WarpingLCSS to spot activities. With the uniformity in extracting activity-specific patterns from raw sensor signals, our proposed system is compatible with respect to modalities and body-worn positions.

We evaluate our system on the Opportunity dataset of four subjects consisting of 17 hard-to-classify classes (e.g., open/close drawers at different heights) with 17 sensors belonging to three modalities (accelerometer, gyroscope and magnetic field) attached at different on-body positions. The system achieves good performances (63% to 84% in F1 score). Moreover, the robustness and efficiency to addition and removal of sensors as well as activity classes are also investigated.

Author Keywords
Continuous Activity Recognition; Multi-modality; Multi-sensor; Body-worn Sensors; WarpingLCSS

ACM Classification Keywords
H.1.2 [User/Machine Systems]; I.5.2 [Pattern Recognition]

Introduction
Continuous activity recognition (*activity spotting*) is a core component in context-aware systems. It enables a variety of applications such as ambient assisted living, human computer interaction. In activity spotting, actions of interest and their temporal boundaries are detected in a continuous data stream in which they are randomly mixed with arbitrary non-interest actions (*null* class).

In the past few years, promising results from body-worn sensors for activity spotting have been presented [1, 13]. Many modalities, such as motion-related ones (acceleration, rate of turn, magnetic field) [1, 5], temperature [6] or sound [7] have been explored as inputs to activity recognition systems. Nevertheless, with the increasing availability of commercial wearable sensor devices (such as smartphones, watches, glasses and, in a near future, sensor-equipped garments), the multimodal aspect is ready for being fully exploited.

Why are there no multimodal activity spotting systems readily deployed in commercial applications? One challenge is that recognition chains for different sensor modalities need still quite some hand-crafting for being deployed. Features and classifiers used for accelerometer data differ substantially from the ones needed for gyroscope or compass data, or audio. Even for the same modality, the needed features can differ for sensors mounted on different parts of the body. Furthermore, since each modality allows to recognize some restricted sets of activities, system designers still tend to solve specific activity recognition problems with specific sensors.

Other challenges of a unified multimodal framework that make it easily deployed in any settings are the following:

- If new sensors are worn by the user (e.g., the user wears a new smart watch), these should be integrated into the system smoothly, without asking the user where on the body the devices have been mounted or which classifier should be used for new data from those sensors.
- The system should handle missing sensors in run-time (e.g., the user gets off his sensor-equipped shoes) without interfering with other sensors.
- The system should also adapt new activity classes of interest smoothly without retraining the whole system.

In this paper we make one step towards a unified framework for multimodal activity recognition which attempts to overcome the challenges addressed above. Our system treats different modalities, and sensors with the same modalities at different on-body placements in a homogeneous way. Specifically, each data stream is first quantized into strings of symbols by using k-means (here serving as a vector quantization step) and then substring matching is performed by the fast and efficient template matching method WarpingLCSS [8] to spot activities. Our system recognizes activities by extracting activity-specific patterns from raw sensor signals, hence it is agnostic with respect to modalities and on-body positions. Thus, it can combine any available wearable sensors placed on user's body to spot activities. Besides that, our system takes the benefit of template matching methods in which different

classes can be trained and spotted separately so that it can handle new activity classes easily.

We investigate two multimodal frameworks to fuse different data sources either at the signal level (*signal fusion*) or at a decision level (*classifier fusion*). In the classifier fusion framework, a novel fusion technique for template matching is proposed to combine all spotting results from different sensors. The two frameworks will be compared throughout the paper in terms of recognition performance, speed, ease to add or remove sensors, and ease to add or remove activity classes.

In this work, we test the proposed frameworks with the recognition of hand actions (i.e., gestures), but the frameworks apply with no loss of generality to other activities. The proposed system is evaluated with the complex Opportunity dataset [10], which includes 17 hand actions using 17sensors belonging to three modalities (accelerometer, gyroscope and magnetic field) at different on-body positions. The performances of all subsets of sensors in the *classifier fusion* framework are also given to demonstrate its flexibility to sensor addition and removal.

Related Work

Various techniques for online gesture recognition can be found in literature; they include Hidden Markov Models (HMM) [12], support vector machines (SVMs) [14], template matching methods (TMM) using dynamic time warping (DTW) [13] or using longest common subsequence (LCSS) [8].

With recent advances in the development of inexpensive wearable sensors, researchers have investigated activity recognition systems using multimodal sensors or multiple single-modal sensors to improve the performance. In [1], five accelerometers were attached at different on-body positions to recognize physical activities. [9] used motion sensors and force sensing resistors to recognize hand actions for quality inspection in car production. The fusion of multiple data sources can be performed either early at signal level, feature level [6], or late at decision level (i.e., classifier fusion) [1, 9].

For each modality, a wide range of features and supervised learning techniques for activity recognition has been explored [4]. As one example, accelerometer data can be classified with Naive Bayes [1], SVMs [14], C4.5 decision trees [1], HMM [12], TMM [8] and a variety of features in both time and frequency domains can be extracted [4]. Due to the diversity of methods and features, the existing multimodal systems selected different methods and features for different modalities or sensors mounted at different on-body positions [9, 13]. For example, in the application of car quality inspection [9], inertial sensors must be attached at arms and torso in order to acquire the trajectories of wrists and elbows, force sensing resistors attached at lower arms to monitor muscle. Four different methods including K-Nearest-Neighbor (KNN), TMM, k-means classifier, and Bayes classification were used in that work. They also trained different classifiers linked to the different modalities with different set of labels for the best possible performance purpose.

WarpingLCSS was first presented in our previous work [8] as a fast and efficient method to spot gestures using one 3D accelerometer on arm. The method showed robustness against noisy annotations and high variances in activity execution. In the work by Chen et al. [2], they investigated WarpingLCSS with multi-sensor fusion combining 6 different accelerometers at wrists and arms. However, the performance was not improved. According to the best of our knowledge, there is no previous work

that investigates the use of WarpingLCSS towards a unified multimodal system, especially with other modalities such as gyroscopes, magnetic sensors.

Multi-modality System

In our system, we use the recently proposed template matching WarpingLCSS [8] as a core module for data processing, training and activity spotting. Data is recorded continuously and synchronously from multiple sensors. The training data is manually labeled with a list of activity classes of interest. Activities which are not in the list are considered as *null* class.

We propose two frameworks for fusing multimodal sensors at two different processing levels: *classifier fusion* and *signal fusion*. In the *classifier fusion* framework, signal data from each sensor are processed separately by a template matching (TM) module and then the spotting outputs from all sensors will be fused to have a final recognition output. Different TM sessions can be run in parallel. In contrast, in the signal fusion framework, signals from all sensor modalities are combined into one data stream before being fed into the TM module. Figures 2 and 4 gives an overview of the two frameworks.

Template Matching

The TM module processes input data, generates templates for activity classes in the training phase and recognizes activities in the spotting phase. An overview of the TM module is shown in Figure 1.

First, the TM module applies k-means to all training data points and quantizes the signal input to their closest cluster centroids. Thus, signal data are then represented as a string of symbols (i.e., the indices of the centroids). The number of symbols k depends on the variation of the input signal. Accordingly, the cluster centroids are representative points which can capture body movements at sensor-attached positions regarding to specific activities.

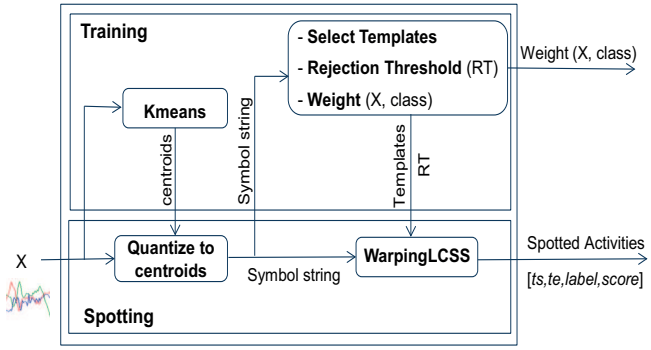

Figure 1: Template Matching Module. X is data input from one or multiple combined sensors.

In the training process, one or more templates are created for each activity of interest to represent the typical patterns for that class. The templates are chosen as instances that have the highest average longest common subsequence (LCSS) scores [3] to all other instances of the same class. Additionally, a rejection threshold needs to be calculated for each activity class in the training phase to be able to reject signals not belonging to that class upon recognition. Let $\mu^{(c)}$ and and $\sigma^{(c)}$ be the mean and the standard deviation, respectively, of LCSS values between the template of a class c and any string belonging to the same class. We calculate the rejection threshold to be below $\mu^{(c)}$ by some standard deviations: $\mu^{(c)} - h^{(c)} * \sigma^{(c)}$, with $h^{(c)} = 0,1,2,....$ The value of $h^{(c)}$ is determined by testing the recognition of class c on the training data and selected as the one which yields the best F1 score performance. Specifically,

$$F1_c = 2 * \frac{precision_c * recall_c}{precision_c + recall_c}, \quad (1)$$

where $precision_c$ is the proportion of samples of class c predicted correctly over the total samples predicted as class c; $recall_c$ is the proportion of samples of class c predicted correctly over the total samples of class c. Note that the value of $F1_c$ can also be used to indicate how well sensor data fed into the TM module can recognize the specific class c.

When spotting, the same process of quantization is applied to the streaming sensor data, with the cluster centroids identified during training. Then, for each activity class, the WarpingLCSS method [8] is used to match the template within the online string to spot activities belonging to that class. Given the activity template for class c, $\bar{s}^{(c)}$, the WarpingLCSS score $W_{(\bar{s}^{(c)},s)}(i,j)$ between the first i symbols of the template $\bar{s}^{(c)}$ and the first j symbols of the string s is obtained as follows:

$$W_{(\bar{s}^{(c)},s)}(i,j) = \begin{cases} 0 & \text{, if } i=0 \text{ or } j=0 \\ W_{(\bar{s}^{(c)},s)}(i-1,j-1)+1 & \text{, if } \bar{s}^{(c)}(i)=s(j) \\ \max \begin{cases} W_{(\bar{s}^{(c)},s)}(i,j) - p * d(\bar{s}^{(c)}(i), s(j)) \\ W_{(\bar{s}^{(c)},s)}(i-1,j) - p * d(\bar{s}^{(c)}(i), \bar{s}^{(c)}(i-1)) \\ W_{(\bar{s}^{(c)},s)}(i,j-1) - p * d(s(j), s(j-1)) \end{cases} & \text{, otherwise,} \end{cases}$$

where p is a penalty parameter of the dissimilarity and $d(\cdot,\cdot)$ is the normalized Euclidean distance between two symbols (i.e., two corresponding centroids) in a range [0,1] . When a new symbol arrives, the WarpingLCSS processes and updates the score immediately. Hence the computational cost of WarpingLCSS is low for online recognition. The WarpingLCSS score grows (i.e., symbols are matched) when an instance of the examined class is performed and drops significantly if other classes are performed due to the penalty terms. Additionally, the penalty is accumulated in a time-warping manner as in dynamic time warping (DTW) [11], hence WarpingLCSS penalizes the same consecutive symbols which are mismatched only once. An activity of class c is recognized for each local maximum of W that exceeds the rejection threshold.

Outputs of the TM module are spotted activities with a format *[start-time, end-time, label, simScore]* to indicate when the activity occurs and the similarity score (*simScore*) between the activity and the template of that class. In the TM module, the spotting of different activity classes can be processed concurrently in parallel. A more detailed explanation, complexity analysis and illustration of WarpingLCSS can be found in [8].

Classifier Fusion Framework

In the *classifier fusion* framework, each sensor is treated uniformly via the same process in the *Template Matching* module. The spotting outputs from all sensors are combined in the *Classifier Fusion* module as shown in Figure 2. Then the *Decision Making* (DM) module resolves conflicts for spotted instances belonging to multiple classes and output the results.

Let Φ and $|\Phi|$ be the set of sensors and the number of sensors in the system, respectively. We represent the spotting output from a sensor $\mathcal{S} \in \Phi$ in a spotting matrix $\mathcal{M}(\mathcal{S})$ of size $\mathcal{C} * \mathcal{N}$, with $\mathcal{C}$ is the number of activity classes of interest and $\mathcal{N}$ is the number of samples processed. $\mathcal{M}(\mathcal{S})(c,i)$ represents the entry at the ith sample and the row of class c in the matrix $\mathcal{M}(\mathcal{S})$. Each row c in the matrix, indicated as $\mathcal{M}(\mathcal{S})(c)$ stores the information of spotted instances of an activity class c from the sensor $\mathcal{S}$. Specifically, if the sensor outputs an activity instance of class c from *start-time* to *end-time* with a similarity score *simScore* (i.e., *[start-time, end-time, c, simScore]*), then $\mathcal{M}(\mathcal{S})(c,i) = simScore$ for all i-th samples in the interval from *start-time* to *end-time* at the

row c. Figure 3 gives an example of the spotting matrix.

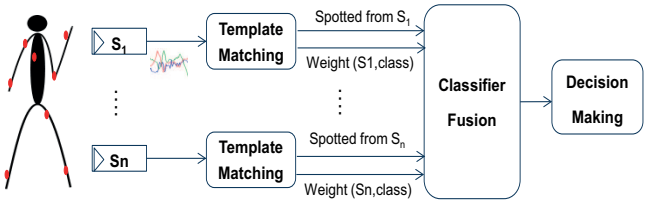

Figure 2: Classifier fusion framework

samples

Activity class	1	2	3	4	5	6	7	8	9	10	...
drink	0	0.8	0.8	0.8	0	0	0	0	0	0	...
open_door	0	0	0	0	0	0	0.6	0.6	0.6	0	...
close_door	0	0	0	0	0	0	0	0	0	0	...

Figure 3: An example of the spotting matrix with three activity classes (drink, open door and close door) and two spotted activities: [2, 4, drink, 0.8] and [7, 9, open_door, 0.6].

Classifier Fusion We propose Weighted Fusion method to fuse the spotting results from all sensors. Let Weight($\mathcal{S}$,c) be a prior weight to indicate how well sensor $\mathcal{S}$ can recognize the specific class c. We set Weight($\mathcal{S}$,c) as the best $F1_c$ performance (see Equation 1) when selecting the rejection threshold for activity class c in the processing of sensor s. The weighted summed spotting matrix is computed as follows.

$$\bar{\mathcal{M}}(c) = \sum_{\substack{i=1 \\ \mathcal{S}_i \in \Phi}}^{|\Phi|} Weight(\mathcal{S}_i, c) * \mathcal{M}(\mathcal{S}_i)(c) \, \forall c. \qquad (2)$$

The similarity score of an activity in the spotting matrix degrades if the prior performance of the sensor to recognize the corresponding activity class is low.

Given the fused spotting matrix $\bar{\mathcal{M}}$, for each spotted activity *[t1, t2, c, simScore]*, the similarity score *simScore* is updated as the average score in the interval from the time t1 to the time t2 at the row c in $\bar{\mathcal{M}}$. Specifically, the updated *simScore* is computed as follows.

$$\bar{simScore} = \frac{\sum_{i=t1}^{t2} \bar{\mathcal{M}}(c,i)}{(t2-t1) \text{ [samples]}}. \qquad (3)$$

Consequently, the similarity score of an activity is boosted if more sensors predict that activity performed.

Decision Making If an activity is spotted as belonging to multiple classes (i.e., boundaries of spotted instances are overlapping), the DM module will resolve conflicts by deciding the class with highest similarity score as the best match. If an activity is classified into only one class, the DM will output the class. Otherwise, if no activity class is spotted, the DM will output *null*.

Signal Fusion Framework

In the *signal fusion* framework, the *Signal Fusion* module combines signals from all sensors into one data stream as shown in Figure 4.

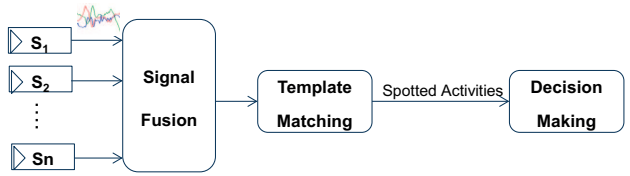

Figure 4: Signal fusion framework

Let d_i be the dimension of signal data generated from sensor $\mathcal{S}_i \in \Phi$. The combined data stream from the *Signal Fusion* module has a dimension of $\sum_{i=1,\mathcal{S}_i \in \Phi}^{|\Phi|} d_i$. The TM

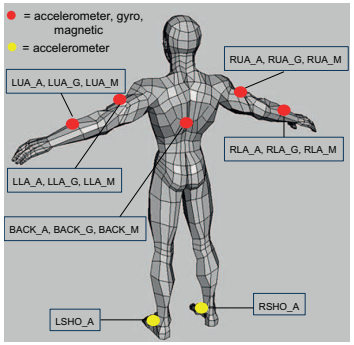

Figure 5: Sensors attached at different places on body and their modalities in Opportunity dataset.

module then processes data and outputs spotted activities. Finally, the DM module handles spotting conflicts and outputs recognized activities as discussed above.

Experiments

We present the activity dataset, evaluation metrics and the conducted experiments to evaluate the proposed system in this section.

Dataset

We evaluate the system on the Opportunity dataset [1] [10] which is a rich multimodal multi-sensor dataset collected in a naturalistic environment akin to an apartment, where users execute 17 daily gestures. The dataset contains a large variability in the execution of the activities and *null* class is predominant (37%). We use the subset of recording corresponding to four subjects in which each subject wears 17 sensors belonging to three modalities (3D accelerometer, 3D gyroscope and 3D magnetic field) attached at different on-body positions. Each subject performs 20-40 repetitions of each gesture class. Totally, the dataset contains 1485 activity instances. Table 1 shows the list of activity classes in the Opportunity dataset. Note that there are three drawers located at different heights and two different doors in the dataset. Figure 5 shows locations of sensors on body (i.e., right upper arm (RUA), right lower arm (RLA), left upper arm (LUA), left lower arm (LLA), back (BACK), right shoe (RSHO) and left shoe (LSHO)) and their modalities. The signals of all sensors are recorded at a frequency of 30Hz.

Null	clean Table (CT)	open Drawer 1-2-3 (ODr1-2-3)
close Drawer 1-2-3 (CDr1-2-3)	open Door 1-2 (OD1-2)	close Door 1-2 (CD1-2)
open Fridge (OF)	close Fridge (CF)	drink Cup (D)
open Dishwasher (ODi)	close Dishwasher (CDi)	Toggle Switch (TS)

Table 1: Activities in Opportunity dataset.

[1] http://www.opportunity-project.eu/challengeDownload

Evaluation Metrics

Generally, activity classes may occur non-uniformly in real-life datasets. In the Opportunity dataset, *null* class is predominant (37%). Therefore, we use the weighted average sample-based F1 score to assess the performance of activity recognition. It is computed as the sum of the F1 scores of all classes, each weighted according to the proportion of samples in that particular class. Specifically,

$$F1 = \sum_{c} w_c * F1_c,$$

where c is the class index, w_c is the proportion of samples of class c, and $F1_c$ is computed as in Equation 1.

We present two ways of computing the F1 score, either including (F1-Null) or excluding the *null class* (F1-NoNull). F1-NoNull does not consider the *null* class, but still takes into account false predictions of gesture samples or instances misclassified as *null class* or vice versa. F1-NoNull value represents how well the recognition system detects activity classes of interest. The recognition system that has high values of both F1-Null and F1-NoNull predicts well both activities and *null* class.

Experiments on Multimodal System

For each subject, we perform experiments in 5-fold cross validation. All raw signals (30Hz sampling rate) are down-sampled for a faster computation. Specifically, an average value of each sliding window of size 6 samples and overlap 3 is extracted to represent the corresponding set of data points in the window. In our experiments, the TM module generates only one template for each activity class.

In the *classifier fusion* framework, the number of symbols (i.e., number of clusters in k-means) is selected empirically k = 20 for each 3D sensor. Note that k can be selected by using cross-validation on the training data.

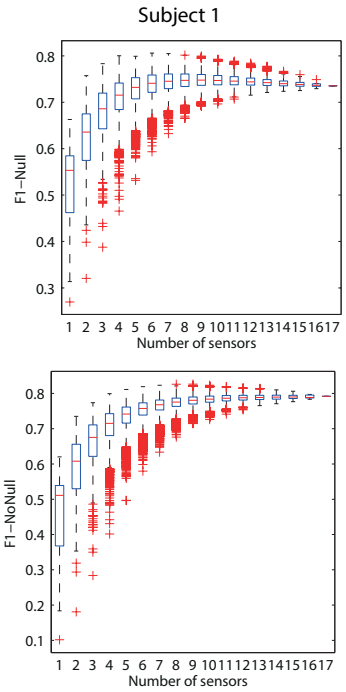

Figure 6: Performance of classifier fusion framework with all subset combinations of 17 sensors for subject 1. The red middle line shows the average performance.

In the *signal fusion* framework, the *Signal Fusion* module combines all 17 sensors into a data stream with a high dimension of 51. Consequently, the number of symbols is selected much higher to capture variants in the combined movements at seven on-body positions (see Figure 5). We select empirically k = 200.

Results and Discussion

Performance of One Sensor

The performance of a sensor reflects how well that sensor recognizes the activities. Figure 6 shows the performance of each sensor (i.e., number of sensors = 1) and their combinations in the *classifier fusion* framework. Due to a space limit, we report only the result from subject 1, however, those performances for other subjects have a similar trend. As seen in Figure 6, the performances of different sensors vary significantly. The sensors on shoes (LSHO_A and RSHO_A) give the worst performance since their signals are not distinguishable for different gesture executions (e.g., open doors and open drawers have the similar patterns of foot movements). The accelerometer at lower dominant arm (RLA_A) gives the best performance for subject 1. Meanwhile, the magnetic sensors at lower dominant arm give the best results for subjects 2-4.

Comparison between Two Frameworks

Table 2 show the results on the use of all 17 sensors in the two proposed fusion frameworks. They both achieve a good performance for the four subjects (63% to 84% F1-Null). In average, the performance of the *classifier fusion* framework on 17 sensors increases by 16% F1-Null and 21% F1-NoNull compared with the average performance of one sensor. It also increases by 4% F1-Null and 15% F1-NoNull compared against the average performance of the best one-sensor.

The *signal fusion* outperforms the *classifier fusion* about 7% F1-Null and only 1% F1-NoNull in average. It means the *signal fusion* can detect the *null* class better than the *classifier fusion*. The rationale is that the *signal fusion* has a global view of data from all sensors at once before processing; meanwhile the *classifier fusion* framework has only a local view of data from each sensor. The hand actions of concern and the *null* class may have the similar foot movements (e.g., walking, standing). Hence, data from the shoe sensor may detect the activities when the *null* class actually occurs. Even the other sensors can detect the *null* instance, the classifier fusion still outputs the false detected activities. By contrast, the *signal fusion* framework outputs an activity only when the combined pattern of that activity from different sensors is matched.

Besides the recognition accuracy, we compare the advantages and limitations of the two frameworks with regards to speed, ease to remove or add sensors to the system, and ease to add or remove activity classes. They are summarized Table 3.

	Classifier fusion	Signal fusion
Parallelism at sensor	* Yes	* Do not care
Parallelism at activity class	* Yes	* Yes
Sensor Addition or Removal	* Easy	* Hard
Class Addition or Removal	* Easy	* Easy

Table 3: Comparison between classifier fusion framework and signal fusion framework.

The running time of our proposed system depends on how many sessions of the TM module run to spot activities. In the *signal fusion* framework, the number of TM session is only one. In the *classifier fusion* framework, it equivalents to the number of sensors deployed in the system. However, those TM sessions for different sensors can be executed in parallel. Therefore, if the parallelism is

	Subject 1		Subject 2		Subject 3		Subject 4		Average	
Method	F1-Null	F1-NoNull	F1-Null	F1-NoNull	F1-Null	F1-NoNull	F1-Null	F1-NoNull	F1-Null	F1-NoNull
Classifier Fusion	0.74	0.79	0.63	0.67	0.75	0.80	0.65	0.71	0.69	0.74
Signal Fusion	0.77	0.77	0.67	0.68	0.84	0.81	0.74	0.73	0.76	0.75

Table 2: Performance of two frameworks on 17 sensors in the Opportunity dataset.

maximized, the running time of the two frameworks to spot activities is equivalent (i.e., time to run the TM module once).

Each sensor is processed separately in a uniform way in the *classifier fusion* framework. Therefore, it enables easy addition and removal of sensors without interfering with the use of other sensors to spot activities. Meanwhile, in the *signal fusion* framework, the combination of all sensor signals into one data stream before being processed will require the same sensor settings in the training and spotting phases so that the quantization step can proceed properly. Hence, adding or removing sensors in the *signal fusion* framework requires retraining the whole system.

The TM module spots each activity class separately. Therefore, the spotting for different activity classes can be executed independently in parallel. Hence, our proposed recognition system with the core *Template Matching* is very flexible in adding or removing activity classes.

Sensor Combinations in Classifier Fusion Framework
We show the performances of the classifier fusion framework on all subset combinations of 17 sensors in Figure 6. The results show that the performances among groups of sensors differ less when the number of sensors increase. This indicates that adding a better performance sensor into a group increases the average performance. The average performance increases significantly in both F1-Null and F1-NoNull as the number of sensors increases from 1 to 6 and then keeps stable. Generally, the combination of more sensors does not always yield the better performance (e.g., two accelerometers at lower arm and upper arm may not improve the detection of OD1 and OD2). The best-combination performance increases dramatically as the group size increases from 1 to 4. F1-NoNull is almost unchanged after reaching the best performance. Meanwhile, the F1-Null of 17 sensors is less than the best performance by 10% in average. As discussed above, the presence of not-so-distinguishable sensors (e.g., shoe sensors) in the *classifier fusion* makes the recognition more confused in detecting the *null* class.

Table 4 gives the subset combination of 17 sensors that gets the best result. The orientation sensors on the back and arm (gyroscope and magnetic sensors) distinguish well the hard-to-classify activities in the Opportunity dataset.

	Sensors (Number of sensors)	F1-Null	F1-NoNull
Subject 1	BACK_M, RUA_G, RLA_G, RLA_M, LUA_A, LLA_A (6)	0.82	0.83
Subject 2	BACK_G, BACK_M, RUA_G, RUA_M, RLA_G, LUA_A (6)	0.71	0.73
Subject 3	BACK_G, RUA_M, RLA_G, RLA_M (4)	0.87	0.85
Subject 4	BACK_M, RUA_G, RLA_A, RLA_M (4)	0.75	0.74

Table 4: The combination of sensors giving the best performance in the classifier fusion framework.

Conclusion and Future Work

We have introduced the unified multimodal system for activity spotting by processing different sensors in a homogeneous way based on the template matching WarpingLCSS. Two fusion frameworks are investigated:

the classifier fusion and the signal fusion. The results of the experiments show the flexibility and efficiency of our system in handling multimodal sensors. The more sensors are added into the system, the equal or better performance is achieved in average. Moreover, the system is flexible in adding or removing activity classes. The classifier fusion framework provides the ease to add or remove sensors. Meanwhile, the signal fusion framework yields the better performance in classifying *null* classes due to a global view of data. In future, we plan to apply sensor selection algorithms in the classifier fusion framework to achieve the best performance. We also plan to investigate other modalities in our system.

Acknowledgments

This work has been supported by the Swiss Hasler Foundation project Smart-DAYS.

References

[1] Bao, L., and Intille, S. S. Activity recognition from user-annotated acceleration data. In *Pervasive Computing: Proc. of the 2nd Int'l Conference* (2004).

[2] Chen, C., and Shen, H. Improving online gesture recognition with WarpingLCSS by multi-sensor fusion. In *Computer Engineering and Networking*, Lecture Notes in Electrical Engineering. 2014.

[3] Cormen, T. H., Stein, C., Rivest, R. L., and Leiserson, C. E. *Introduction to Algorithms*. 2001.

[4] Huynh, D. T. G. *Human Activity Recognition with Wearable Sensors*. PhD thesis, TU Darmstadt, 2008.

[5] Kunze, K., Bahle, G., Lukowicz, P., and Partridge, K. Can magnetic field sensors replace gyroscopes in wearable sensing applications? In *International Symposium on Wearable Computers (ISWC)* (2010).

[6] Lee, Y.-S., and Cho, S.-B. Recognizing multi-modal sensor signals using evolutionary learning of dynamic bayesian networks. *Pattern Analysis and Applications* (2012).

[7] Nguyen-Dinh, L.-V., Blanke, U., and Tröster, G. Towards scalable activity recognition: Adapting zero-effort crowdsourced acoustic models. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, MUM (2013).

[8] Nguyen-Dinh, L.-V., Roggen, D., Calatroni, A., and Tröster, G. Improving online gesture recognition with template matching methods in accelerometer data. In *the 12th International Conference on Intelligent Systems Design and Applications, ISDA* (2012).

[9] Ogris, G., Stiefmeier, T., Lukowicz, P., and Troster, G. Using a complex multi-modal on-body sensor system for activity spotting. In *IEEE International Symposium on Wearable Computers*, ISWC (2008).

[10] Roggen, D., and et. al. Collecting complex activity data sets in highly rich networked sensor environments. In *7th Int. Conf. on Networked Sensing Systems* (2010).

[11] Sakoe, H. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing 26* (1978), 43–49.

[12] Schlömer, T., Poppinga, B., Henze, N., and Boll, S. Gesture recognition with a Wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction* (2008).

[13] Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., and Tröster, G. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing Magazine* (2008).

[14] Wu, J., Pan, G., Zhang, D., Qi, G., and Li, S. Gesture recognition with a 3-D accelerometer. In *Ubiquitous Intelligence and Computing*. 2009.