Automatic Detection of Urban Features from Wheelchair Users' Movements

Gabriele Civitarese, Sergio Mascetti, Alberto Butifar, Claudio Bettini
Università degli Studi di Milano
Dept. of Computer Science, EveryWare Lab
Milan, Italy
{gabriele.civitarese, sergio.mascetti, claudio.bettini}@unimi.it
alberto.butifar@studenti.unimi.it

***Abstract*—Providing city navigation instructions to people with motion disabilities requires the knowledge of urban features like curb ramps, steps or other obstacles along the way. Since these urban features are not available from maps and change in time, crowdsourcing this information from end-users is a scalable and promising solution. Our preliminary study on wheelchair users shows that an automatic crowdsourcing mechanism is needed, avoiding users' involvement.**

In this contribution we present a solution to crowdsource urban features from inertial sensors installed on a wheelchair. Activity recognition techniques based on decision trees are used to process the sensors data stream. Experimental results, conducted with data acquired from 10 real wheelchair users navigating in an outdoor environment show that our solution is effective in detecting urban features with precision around 0.9, while it is less reliable when classifying some fine-grained urban feature characteristics, like a step height. The experimental results also present our investigation aimed at identifying the best parameters for the given problem, which include number, position and type of inertial sensors, classifier type, segmentation parameters, etc.

***Index Terms*—Motion disabilities, activity recognition, urban navigation**

I. Introduction

Modern navigation systems compute the route depending on the user's current mean of transport. These systems can compute the route for users moving by cars, public transportation, foot and others. However, there is no specific support for users with limited mobility, like wheelchair users. To the best of our knowledge, only *Google Maps* provides some support in terms of information about accessibility in public transportation and limited to some cities[1]. While this is surely a useful service for some users, it addresses a small part of the overall mobility problem by wheelchair users [1].

A major problem, that emerged during an interview with five wheelchair users living in Milan (Italy), is that a person moving on a wheelchair does not know in advance which obstacles she/he will face along a route. For example, even if curb ramps are commonly available at intersections, sometimes they can be missing or occluded by road work. According to interviewed users, these problems are so frequent and their effects are so frustrating that they declare to be reluctant to move along unknown routes.

[1]https://goo.gl/f5BeFc

To address these issues we are currently developing the *Moving Wheels* navigation system that aims at supporting mobility of people with disabilities by guiding them on routes that are personalized according to their needs. For example, consider a user with an electric wheelchair that is unable to climb steps; for this user, the system will compute routes where curb ramps are available.

While the problem of offering personalized navigation instructions to people with disabilities has been addressed before in the literature (see e.g., [2], [3]), and can be adapted to this application, a critical challenge in *Moving Wheels* is to acquire detailed information about *urban features* in the form of obstacles (e.g., a step) and accessibility elements (e.g., a curb ramp). This paper focuses on this challenge and presents a solution to automatically recognize urban features from the movement of the users themselves: while a user moves in the environment on a wheelchair (e.g., climbs up a ramp) her/his mobile device acquires movement information from inertial sensors and automatically detects the urban feature (i.e., the curb ramp). As future work, we intend to design a system to automatically share this information so that it can be used when computing the route for other users. This is a form of data crowdsourcing that does not require user intervention.

This paper has three main contributions. First, it describes *Moving Wheels*, focusing on the definition of the novel and challenging problem of detecting urban features from wheelchair movements. Second, it illustrates the technical solution to recognize urban features that includes data acquisition, labeling, features extraction, and classification with a supervised machine learning technique. Third, the paper presents the results of an extensive experimental evaluation of the proposed technique based on data acquired from wheelchair users in an outdoor environment. Results show which algorithms and which parameters provide the best accuracy, but most importantly they validate the solution with recognition rates higher than 80% for several important urban features.

II. The *Moving Wheels* system

Moving Wheels is a context-aware assistive navigation system being developed by the EveryWare Lab in Milan with two main objectives: first, to provide navigation instructions to people with disabilities guiding them along routes that are personalized according to their needs. To compute these routes, *Moving Wheels* needs detailed information not only about the road network but also about the urban features that may be an obstacle for the navigation. The acquisition of this information is the second objective of *Moving Wheels*.

A mobile client, similar to a traditional navigation app, guides end-users from a start position (generally their current position) to a target destination with a main difference with respect to other solutions: it allows end-users to finely tune preferences with respect to classes of urban features depending on their (dis)abilities. For each class, the user can specify whether the urban features in that class should be avoided or not. A third option is available as well: "avoid if possible" means that the user is able to deal with that urban feature, but this costs some effort. Consider the following example:

- `small-step-up`: avoid if possible
- `small-step-down`: no problem
- `medium-step-up`: avoid
- `medium-step-down`: avoid if possible

The above preferences capture the fact that the user is not able (or willing) to climb up steps of medium height. Vice versa, descending a short step is not a problem for this user. Also, the user would prefer to avoid to climb down steps of medium height and to climb up short steps.

The *Moving Wheels* service performs the route computations requested by the mobile client. When computing a route, *Moving Wheels* will avoid all urban features marked as "avoid" and will try to balance the route length with the number of urban features marked as "avoid if possible". For example, consider two alternative routes: one is 200m long with one urban feature marked as "avoid if possible" while the other is $1.5km$ with no urban features marked as "avoid if possible". In this case the system will automatically suggest the former route, as it is much shorter. In other cases the system may automatically suggest a slightly longer route, if it has less features marked as "avoid if possible". When there is not a stark difference between two or more routes, the system allows the user to select his/her preferred route.

On the server side *Moving Wheels* represents the road network as a directed graph in which each edge is labeled with the urban features that a user will encounter by moving along that edge, as exemplified in Figure 1.

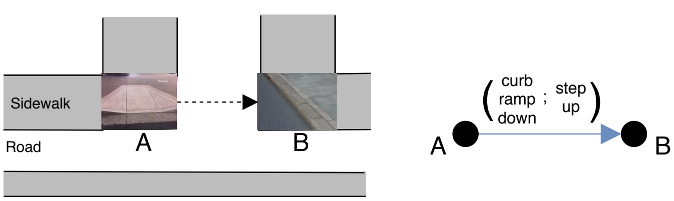


Fig. 1. Road network representation.

A major challenge in *Moving Wheels* is how to acquire the knowledge about the relevant urban features (e.g., steps, ramps) that is needed to populate the graph. We are currently considering these sources:

- existing geo-referenced data stores, including public (e.g., traffic lights from open street map) and private ones (list of curbs ramps from the municipality);
- data annotated by human actors, such as employees, volunteers or end-users, that visit a place either physically or virtually (e.g., looking at *Google street view* images);
- data automatically extracted from geo-spatial image databases (e.g., *Google street view*), adopting computer-vision techniques, similarly to those proposed in [4].

Each of the above solutions has advantages and limitations with respect to a number of factors, including cost (e.g., manual annotation by employees can be costly), scalability (e.g., acquiring data from different municipalities incurs into scalability issues), reliability (e.g, the technique proposed in [4] correctly identifies 93% of zebra crossings), maintenance and types of urban features that can be detected (e.g., some features, like zebra crossings, are easier to detect with computer vision, while others, like the inclination of a curb ramp, are harder to detect).

This contribution focuses on crowdsourcing data from end-users (i.e., people with disabilities). This approach has many advantages: it is scalable, costless, and it keeps information up to date. Since our studies revealed that these users are not really keen to manually enter data or have difficulties doing so, in this paper we show how *Moving wheels* could collect data about urban features by using sensors, without end-user intervention. For example pedestrian crossings can be detected from the camera (e.g., a wearable one) and acoustic traffic signals can be detected from the microphone. We currently focus on urban features that can be detected with inertial sensors mounted on a wheelchair, which include steps, ramps and rough roads.

III. Problem Analysis

During the analysis on the *Moving Wheels* system we conducted informal interviews with two set of users: those using a electric wheelchair and those using a traditional one. The interviews were aimed at better understanding the mobility problems of wheelchair users. We report in the following some observations that are relevant to this contribution.

A. Mobility

There are basically two classes of wheelchairs used for urban mobility[2]: electric wheelchair and traditional ones. The latter can be propelled in three ways: (a) *self-propelled* when the user sitting on the wheelchair use his/her arms to move the wheels, (b) *attendant-propelled*, when a caregiver pushes the wheelchair and (c) *electric-propelled* in which an electric device is attached to the wheelchair to provide motion. Figure 2 shows some examples of wheelchairs.

Generally, electric wheelchairs are used by people who are not able to use a traditional wheelchair (e.g., tetraplegics),

[2]A number of other models are used for indoor use (e.g., in the hospitals), sport and outdoor.

(a) Electric wheelchair. (b) Self-propelled traditional wheelchair. (c) Electric-propelled traditional wheelchair. (d) Sensors position.

Fig. 2. Wheelchairs: types, propulsion modes and sensors position.

while traditional wheelchair are used by people who are able to use a self-propelled traditional wheelchair and that possibly attach an external electric device when really needed.

The ability to move in an urban environment and to face obstacles strongly depends on the wheelchair type, on how it is propelled and on the user's abilities. For example, climbing up a steep ramp is generally not a problem with electric wheelchair, while it can be hard for a self-propelled wheelchair if the user is not well trained. Vice versa, climbing up a step can be impossible with electric wheelchair, while it is generally easier with a electric-propelled traditional wheelchair, or with a self-propelled wheelchair if the user is well trained. In particular users of traditional wheelchair learn to balance themselves on the rear wheels when facing some obstacles: this position, called *balancing*, allows them to climb up and down from steps and is frequently used in other situations in which it is preferable that the front wheels do not touch the ground, like when moving on a rough road.

In this paper we focus on detecting urban features from self-propelled traditional wheelchair. We believe that the methodology and technique we propose in this paper can be easily adapted to the other cases.

B. Sensor data acquisition

Since smartphones include inertial sensors, they could be considered as a data source. For this reason during the interview we asked the participants where they usually keep their smartphone while moving on the wheelchair. It emerges that there are heterogeneous habits: some people using electric wheelchair have a holder (like the tablet in the red circle in Figure 2a), vice versa a common choice among traditional wheelchair users is to store the smartphone in a bag positioned on the rear side of the wheelchair back (like in Figure 2b).

Our preliminary results show that when the smartphone is not firmly attached to the wheelchair frame (e.g., when it is stored in the bag) the collected inertial data is noisy and recognition is harder. For example, consider Figure 3 that shows accelerometer data recorded by a smartphone stored in a bag while the user is moving on a smooth surface. We can observe that, while the user is only accelerating along the frontal direction, spikes are observable on all three axes. This is due to the fact that the bag keeps swinging while the user moves and the smartphone inside the bag moves and rotates in all directions.

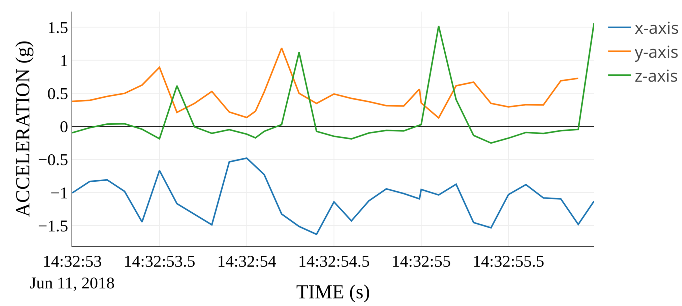


Fig. 3. Sensor data acquired from a smartphone stored in a bag.

For this reason the technique proposed in this contribution is designed to use data acquired from sensors that are attached to the wheelchair frame and whose position and orientation are known. We believe that this setting is realistic, since we expect that smart wheelchairs, equipped with inertial sensors, will become common in the next years [5].

C. The urban features of interest

The main focus of the interviews was to understand the challenges that arise when moving with a wheelchair in a urban environment. The following environmental features emerged to be relevant for wheelchair users:

- steps: their height and whether they should be climbed up or down;
- ramps: their inclination and whether they should be climbed up or down;
- pavement: whether it is flat or inclined (up or down and how much) and whether it is a smooth pavement, asphalt or dirt road;
- bumps and gutters: their height;
- movement aids: like lifts and stairlift.

Based on the observations emerging from the interviews we derived the hierarchical set of labels shown in Figure 4. Each label corresponds to a user's action that discloses the presence of a urban feature. For example `obstacle-step-up-M` indicates that the user climbed up a step of medium height. By knowing the user's position and direction at that time we can recognize the urban feature (the step), its characteristics

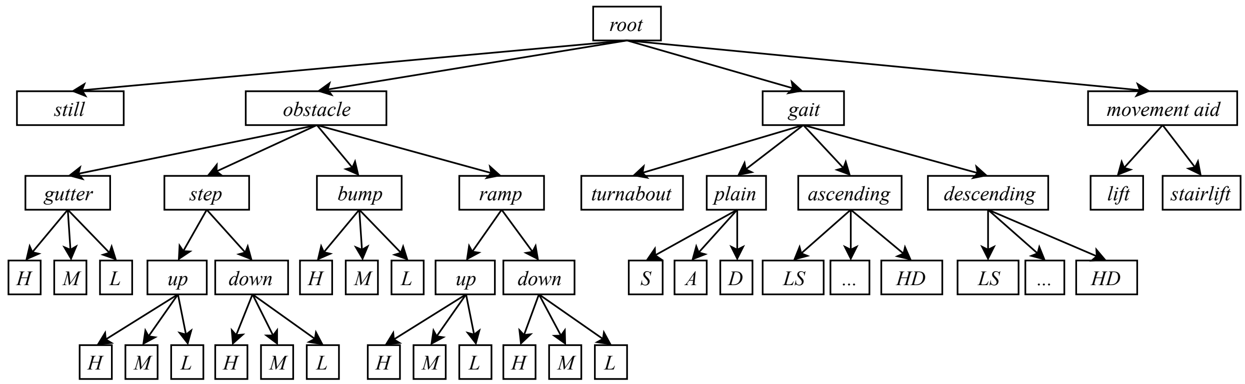


Fig. 4. Labels' hierarchy as emerging from the analysis. H=*High*, M=*Medium*, L=*Low*, S=*Smooth*, A=*Asphalt-like*, D=*Dirt road*, LS=*Low* and *Smooth*, etc.

(medium height) and its orientation (whether it should be climbed up or down when following the same route as the user).

There are two labels that are exceptions as they do not disclose a urban feature: `still` and `turnabout`. The former indicates that the user is not moving, so there is no urban feature to detect. The latter instead does not disclose an exact urban feature but can be used to infer that the user cannot overcome an obstacle and hence can lead to infer a generic accessibility issues when the same behaviour is observed by several users in the same location.

The first level of labels contains: `obstacle`, `gait`, `movement aid` and `still`. Obstacle represents events with a short temporal duration (intuitively between a fraction of a second and few seconds) while the other events have a longer duration. We discretize step, bumps and gutter heights as well as ramps inclination into three classes (high, medium, low). Similarly, we use three classes for surface smoothness: smooth, asphalt-like and dirt road. When the user is moving along an ascending or descending path, we aim at classifying all combinations of inclination (high, medium, low) and of surface smoothness (smooth, asphalt-like and dirt road).

IV. Automatic detection technique

In order to recognize the urban features of interest, we use machine learning techniques adapting to our specific domain an approach widely used for sensor-based human activity recognition. The current implementation of our method relies on batch learning: data are first acquired from wheelchair users, then manually annotated with the ground truth, and finally used to train a supervised classifier. Once the recognition model is trained, our system can detect wheelchair users' actions in real-time.

Section V describes how we collected and annotated sensors data. In the following of this section we describe the main steps of the data management process that enables supervised classification.

A. Data pre-processing

The user's wheelchair is equipped with several devices, placed in different positions with predefined orientation, each acquiring data from various inertial sensors. Data acquired from these sensors is pre-processed in three main steps: data cleaning, fusion and segmentation.

A common technique for data cleaning is data smoothing, which aims at reducing the intrinsic noise of inertial sensor measurements [6]. Many techniques have been adopted in the literature (e.g., median filter). However, in our domain it emerged that data smoothing may actually decrease the recognition rate. We believe that the reason is that some obstacles are crossed in short time and they result in peaks in sensor measurements. Smoothing those peaks removes important information that is needed to correctly detect obstacles. Hence, in our application cleaning has been mostly focused on identifying unreliable sensors and real outlier data.

Data fusion consists in temporally aligning the data streams originated by each sensor. We achieve this by acquiring sensor data from a single gateway (e.g., a smartphone in our case) and timestamping the data with the gateway clock.

After data fusion, sensor data is segmented using a temporal sliding window approach. The application of this method is influenced by two parameters: window temporal length l in seconds, and windows overlap factor o in percentage. Despite being a quite standard and simple segmentation technique, it proved to be effective in our application domain. Figure 5 shows an example of data fusion and segmentation with $l =$ 2sec and $o = 50\%$.

B. Segments labeling

The wheelchair movements (we also called them *activities*) that we need to detect have different duration, from a fraction of a second for *obstacles* to several tens of seconds, for *gait* or *still*. Figure 6 shows an example: a *step* is performed between two *gait* activities.

As usual for supervised learning approaches we faced the problem of how to assign a ground truth label to each segment. A possibility is to use very short segments, so that each one temporally overlaps a single activity. However, as we experimentally show, using very short segments results in poor classification quality. On the other hand, by using longer segments it is possible that a user performs more than one activity during a single segment, as shown in Figure 6 (see the

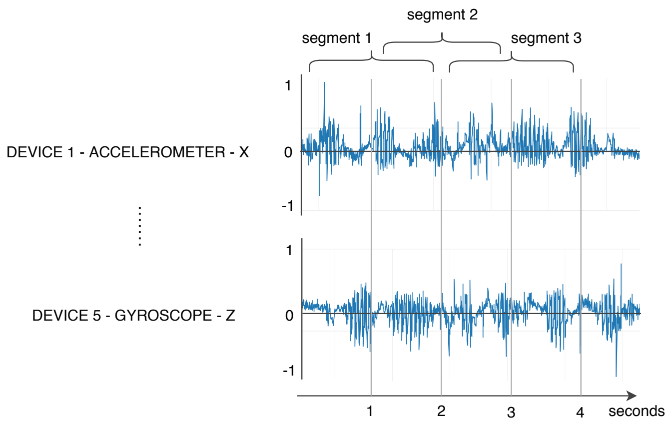


Fig. 5. Segmentation.

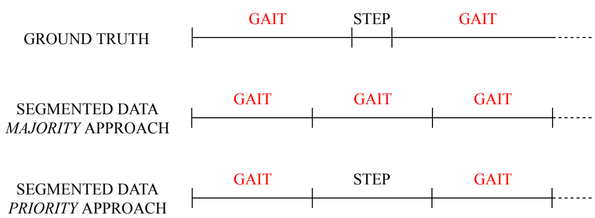


Fig. 6. Labelling approaches.

second segment). In this case a solution is to label a segment according to the prevalent activity for that segment (the one that is performed more than any other during the segment duration). We call this the *majority* approach and an example of its application is shown in Figure 6.

The *majority* approach turned out not to be effective in our domain due to the fact that obstacles are generally crossed in a very short time (e.g., half a second). Indeed, since segments have a length in the order of seconds, none of them is labeled as an obstacle (as in Figure 6). To address this issue we adopt a *priority* labeling approach. The intuition is that obstacles are particularly relevant in our domain, so we give them an higher priority when labeling a segment: if a segment overlaps with an obstacle at least for a given percentage p of the segment length, then we label the segment as *obstacle*, independently from the other labels. This is shown in Figure 6: the second segment has an overlap of 25% with a step (a type of obstacle), so, assuming $p = 25\%$, the segment is labelled with *step*.

C. Feature Extraction

From each segment, we extract several statistical features which are widely adopted in the literature for activity recognition from inertial sensors [7]. In particular we use the following features:

- For each axis of each sensor: *minimum*, *maximum*, *difference between maximum and minimum*, *mean*, *standard deviation*, *variance*, *median*, *root mean square*, *kurtosis*, *skewness*, *zero crossing rate*, *number of peaks* and *energy*;
- For each pair of axis of a given sensor: *Pearson correlation* and *coefficient cross correlation*
- For all axes of a given sensor: *magnitude*.

Overall, we compute 46 features for each 3-axis inertial sensor. For instance, considering that in our experimental setup we used three devices with three 3-axis sensors each, a total of $46 \times 3 \times 3 = 414$ features were used.

D. Supervised Classification

In order to classify urban features, we rely on supervised machine learning algorithms. We experimented with state-of-the-art generative and discriminative classifiers including Multinomial Naive Bayes, SVM, Random Forest and Multi-layer Perceptron. As we show in the experiments, Random Forest resulted to have the highest recognition rate.

Given that our set of labels is naturally represented as a hierarchy, we designed and implemented a hierarchical Random Forest classifier [8]. In this approach a separate classifier is used for each internal node of the hierarchy tree. A segment is first classified by the root classifier as belonging to one of the first level labels (for example it is labeled as *obstacle*), and then considered by a specialized classifier in order to get a label from the second level (for example as *bump*), and further descending the hierarchy until eventually being assigned a label corresponding to a leaf (for example a high bump). We compared this classifier with a *flat* version with experimental results reported in Section VI.

V. Training Data Acquisition

In order to validate our method, we acquired a training set of urban features collected by 10 actual wheelchair users that volunteered to participate in our experiments. In this section we show our experimental setup, the acquisition protocol and we describe the collected dataset.

A. Experimental setup

For the sake of this work, we used MbientLab's *MetaMotionR* devices[3]. A *MetaMotionR* is a multi-sensor board which communicates through Bluetooth Low Energy. Among the several sensors mounted on the board, we use data from accelerometer, gyroscope and magnetometer. As Figure 2d shows, we placed 5 of those devices in different wheelchair's positions: front-left, front-right, rear-left, rear-right and rear-center. Due to technical problems with the devices we collected complete data recordings only from three of them (front-left, rear-right and rear-center). Partial recording from the other two sensors was discarded during data cleaning. Each device streams sensor data to a smartphone at 25Hz. The smartphone stores this information using the MetaBase application[4] assigning to each incoming sensor data the smartphone timestamp. Moreover, we also collected sensor data produced by a different smartphone placed on a bag hanging on the wheelchair back (see Figure 2b). Since the data collected by this second smartphone turned out to be highly noisy, we also ignored this data stream in the following analysis.

[3]https://mbientlab.com/product/metamotionr/
[4]https://mbientlab.com/tutorials/Apps.html

B. Data acquisition

Experiments were conducted at *Spazio Vita*, a Non Profit Organization (NPO) based in Milan, Italy, that supports people with motion disabilities. This NPO owns an outdoor *training area* which includes common urban obstacles, like steps, ascents, etc. The training area is closed to traffic, so that wheelchair users can practice moving in a urban environment without hazards.

Data collection involved 10 users of traditional wheelchairs that self-propelled the wheelchair during the experimental session. For each individual, the data acquisition process includes the following steps: a) take a video where the individual expresses the consent for data acquisition and analysis, b) MetaMotionR are deployed on the wheelchair, c) the MetaBase application on a smartphone is started in order to collect sensor data, d) the user crosses a predefined route while being video recorded. The route consisted in going on a dirt road, going on asphalt, being still, doing turnabout, going up and down on inclined roads with different slopes (high, medium and low), and going up and down on steps with different heights (high, medium and low).

During the data acquisition we noticed a high variability of ways of crossing urban features between different users. For instance, not all users were able to go up or down all steps (e.g., going up a high step is difficult for many users). Moreover, some users cross difficult urban features (like descents, dirt roads and steps) in the balancing position, while others do not. We also noticed that the speed at which wheelchair users cross urban features is highly variable, mainly based on the disability and health conditions of the subject.

For each individual we recorded between 8 and 12 minutes of data. The overall time required to acquire data from each user was between 20 an 30 minutes, mainly due to the time required to deploy the devices on the wheelchair. In the following we call *session* the data acquired from an individual.

C. Data annotation and cleaning

Data annotation was performed offline, thanks to video recordings. In order to synchronize sensor data with video recordings we collected the sensors data from the same smartphone used to record the experiments and we used an application that prints the device time on each frame[5]. Actual data annotation was performed using Anvil [9], a free video annotation tool originally developed for gesture recognition.

As we mentioned above, many activities annotated with a given label are actually performed in different ways (i.e., with different physical movements). In general we expect our system to be robust against variability and indeed we have several examples of actions performed in different ways in our dataset. However we removed a few seconds of data recording in those cases in which a given activity is performed with a physical movement that is not repeated in the dataset. There are just two of these cases: i) a user descended a step while moving backward and, ii) a user climbed up a step while pulling himself with the handrail. Finally, we also excluded from the dataset all the occurrences of the urban feature *step up high*, which was performed only 4 times in the whole dataset due to its intrinsic difficulty.

[5]http://www.timestampcamera.com/

D. Dataset description

Not all urban features identified in Section III-C are available in the environment where we conducted the experiments. In particular the only available obstacles are the steps, while there are no bumps or gutters. There are indeed some ramps, but they are about 8 meters long, so we do not classify them as obstacles, which should take a short time (e.g., a curb ramp is an obstacle) to cross, but instead we classify them as *gait-ascending* or *gait-descending*. The urban features that we have been able to collect are represented in Figure 7.

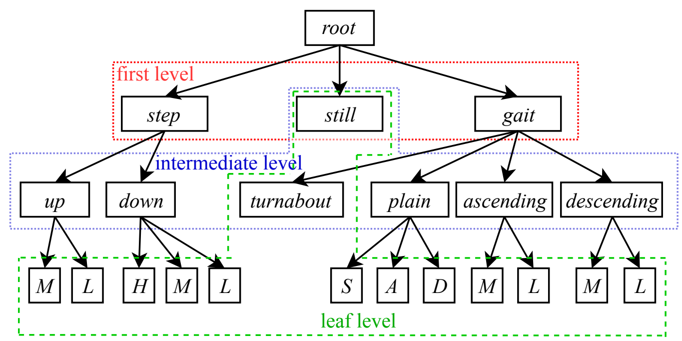


Fig. 7. Hierarchy of labels collected during experiments. H=*High*, M=*Medium*, L=*Low*, S=*Smooth*, A=*Asphalt-like*, D=*Dirt road*.

TABLE I
OVERALL NUMBER OF INSTANCES AND SECONDS OF SENSOR DATA RECORDED FOR EACH URBAN FEATURE

Urban Feature	#instances	#seconds
Step down high	9	8s
Step down medium	18	14s
Step up medium	14	15s
Step up low	34	27s
Step down low	43	31*s*
Gait plain on dirt road	16	218s
Gait descendent medium slope	48	230s
Gait ascendant medium slope	43	248s
Gait descendent low slope	54	252s
Turnabout	119	295s
Gait ascendant low slope	53	304s
Gait plain indoor	27	362s
Stop	63	628s
Gait plain on asphalt-like	368	2821s

Table I shows some details of the collected data. From this table emerges that the dataset is very unbalanced. This is due to the fact that many users were not able to cross specific urban features (e.g. high/medium steps) and those who were actually able could not repeat them several times, as these activities are physically demanding. Moreover, the time required to cross an obstacle like a step is often very short (e.g., half a second).

VI. Experimental evaluation

A. Evaluation methodology

We measure the quality of our urban feature detection system by analyzing the confusion matrix and computing the standard metrics of precision, recall and F1-score. We adopt a leave-one-subject-out cross validation method: at each fold we use nine sessions (one for each individual) to train our model, using the remaining one to test it.

Since we consider a hierarchy of labels, we are interested in investigating the quality of our classifier at different levels of our hierarchy. Indeed, while it would be desirable to accurately detect urban features at the finest granularity (e.g., distinguish a high, medium and low step-down), we are also interested in the recognition rate for coarser-grained urban features, like, for example, whether an obstacle is present or not, or whether a step has been climbed up or down. For this reason we identify three groups of nodes in our hierarchy as shown in Figure 1: *coarse-grained*, *mid-grained* and *fine-grained*.

B. Configuration yielding the best results

We tested various classifiers and a number of parameters trying to identify those yielding the best results. While the results of these tests are illustrated later in this section, the configuration that gave the best overall results is the following:

- a flat Random Forest classifier;
- all available sensor data (3 devices, each with accelerometer, gyroscope and magnetometer);
- a window size of $l = 2$sec and overlap $o = 50\%$;
- a *priority* approach to segments labelling with $p = 20\%$;

In the following of this section we report the results obtained with this configuration, unless differently specified.

Considering the detection of coarse-grained activities (see Table II) the classifier is overall reliable, with particularly good results for the detection of the *gait* and *still* activities. At a finer level (see Table III), results present large differences among the various activities. Indeed, while the classifier can generally correctly distinguish a *step-up* from a *step-down*, it is much less reliable for *gait-ascending* and *gait-descending*. The reason, shown in the confusion matrix reported in Figure 8, is that both labels are frequently classified as *gait-plain*. More generally, from the confusion matrix it emerges that all the activities are sometimes confused with *gait-plain*. We believe that this is due to the fact that the dataset is unbalanced and *gait-plain* has a much larger support with respect to the other activities (see Table I).

TABLE II
Classification rate at the coarse-grained level.

Class	Precision	Recall	F1 score
Obstacle	0.893	0.814	0.851
Gait	0.978	0.986	0.981
Still	0.935	0.912	0.923

Considering fine-grained activities (Table IV), the classifier is only partially reliable. This is due to the fact that, at this granularity, the classifier is often not effective in distinguishing two sibling labels. Consider for example *step-down*: while this label is recognized with high precision and recall, its child *step-down-medium* is not; this is due to the fact that in about 50% of the cases *step-down-medium* is actually classified as *step-down-low*.

TABLE III
Classification rate at the mid-grained level (*still* is omitted)

Class	Precision	Recall	F1 score
Step-up	0.847	0.727	0.783
Step-down	0.892	0.847	0.869
Plain-gait	0.807	0.945	0.871
Gait-ascending	0.833	0.568	0.675
Gait-descending	0.737	0.308	0.434
Gait-turnabout	0.806	0.669	0.731

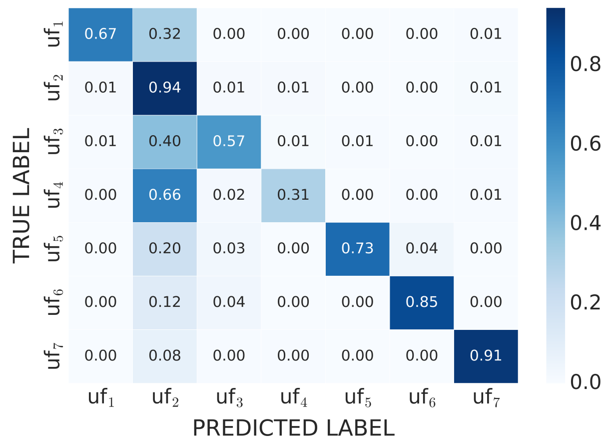


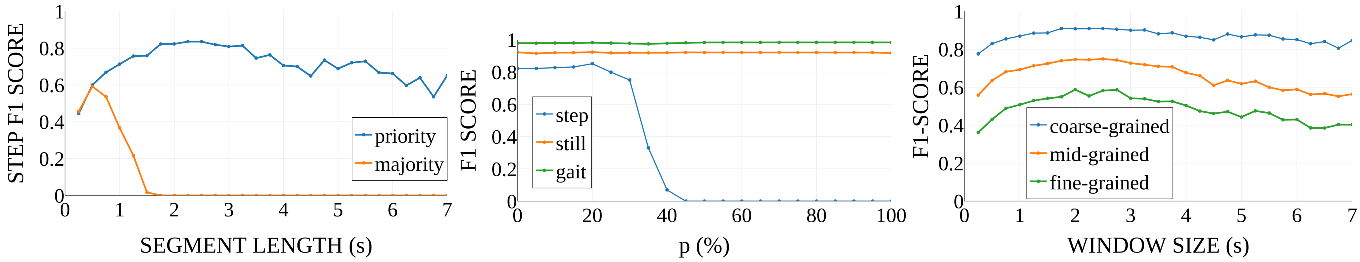
Fig. 8. Confusion matrix at the *mid-grained* level. Indexes: uf_1=*gait-turnabout*, uf_2=*gait-plain*, uf_3=*gait-ascending*, uf_4=*gait-descending*, uf_5=*step-up*, uf_6=*step-down*, uf_7=*still*.

TABLE IV
Classification rate at the fine-grained level (*still* and *gait-turnabout* are omitted)

Class	Precision	Recall	F1 score
Step-up-medium	0.688	0.379	0.489
Step-up-low	0.725	0.806	0.763
Step-down-high	0.737	0.737	0.737
Step-down-medium	0.500	0.371	0.426
Step-down-low	0.647	0.663	0.655
Gait-plain-smooth	0.621	0.254	0.360
Gait-plain-asphalt-like	0.709	0.935	0.806
Gait-plain-dirt-road	0.625	0.320	0.423
Gait-ascending-medium	0.803	0.677	0.735
Gait-ascending-low	0.750	0.408	0.529
Gait-descending-medium	0.590	0.385	0.466
Gait-descending-low	0.931	0.172	0.290

C. Effects of parameters and alternative configurations

As observed in Section IV, we considered two segment labelling techniques. Figure 9a shows a comparison of the *majority* and *priority* labeling techniques for different values



(a) *Majority* vs *priority* segment labeling varying l.

(b) Impact of parameter p.

(c) Impact of window size.

Fig. 9. Effects of parameters and alternative configurations.

of the segment length. We can observe that, for small values of l (in the order of half a second), the two techniques provide almost identical results. For larger values, the performance with the *majority* approach rapidly decreases, because the number of segments labeled as obstacle decreases. Vice versa, with the *priority* approach, segments of about 2s give better performance.

The *priority* approach depends on the parameter p. Figure 9b shows that the detection rate of *still* and *gait* is basically not affected by p. Instead, step detection increases between 0% and 20% and then it rapidly decreases. This is due to the fact that, for larger values of p, the *priority* approach is similar to the *majority* approach.

Figure 9c shows how the recognition rate is affected by the window size l at different levels of our hierarchy. The best results are obtained with $l = 2$. Indeed, for lower values the performance rapidly decreases at all three levels. Instead, for values of l larger than 2 the performance degrades especially for the *mid-grained* and *fine-grained* level.

Our experiments show that the overlap factor o has little impact on the recognition rate at all levels. Slightly better results are achieved with $o = 50\%$ and hence we used this value. Still, since the classifier exhibits similar performance with $o = 0\%$, this value can be also considered as a possible choice, as it minimizes the computational costs because for smaller values of o less segments need to be processed.

Figure 10a shows that the combination of all the available devices (front-left, rear-center and rear-right) gives the highest recognition rate. In case a single device is available, it should be positioned on the rear-center. If two are available, the best performance is obtained by positioning one on the rear-center and the other on the front-left. Indeed this configuration actually gives almost the same results as using three devices.

Considering the contribution of specific sensor types (see Figure 10b), the best results are obtained by using all the three we have considered, i.e., accelerometer, gyroscope and magnetometer. The magnetometer is the least effective when used alone. Instead, by using accelerometer only, the results are only slightly worse than with all the sensors. The same holds for the gyroscope. Still, coupling the magnetometer with the other sensors does actually improve the recognition rate, in particular when it is used together with both accelerometer and gyroscope.

Figure 10c shows the performance of different classifiers. A flat random forest provides the best result, in term of average F1 score, both among coarse and fine grained labels. Given the hierarchical structure of our labels, we expected a hierarchical classifer to outperform the others, but actually hierarchical random forest resulted to have almost the same performance (but slightly worse) than the flat version. The same holds for multinomial naive bayes. Two classifiers provide clearly worse results: support vector machines and multi-layer perceptron. We believe that this is due to the relatively small training set.

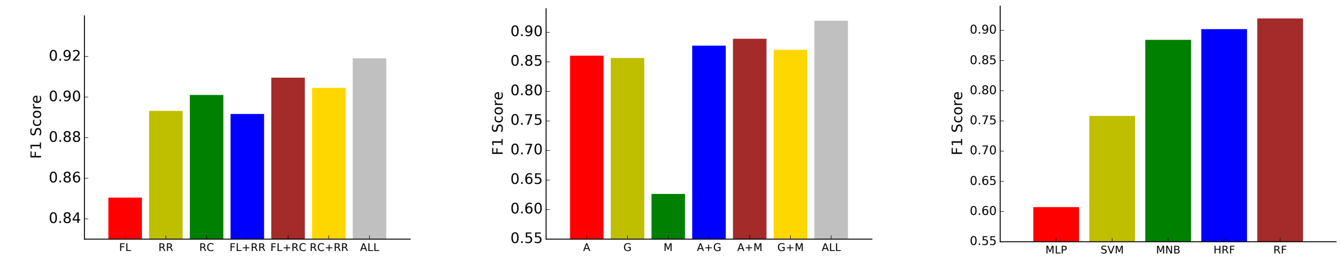
Considering that the training set is highly unbalanced, we expected that a data balancing technique could improve the performance. In particular, we experimented a) Random Forest with balanced class weights and b) well-known techniques to balance the dataset with both oversampling and undersampling [10]. In both cases, unexpectedly, the balancing techniques did not improve the recognition rate.

VII. Related work

A number of commercial solutions have been proposed to detect urban features from images e.g., *Mappillary*[6] or to support people with disabilities during navigation. Similarly to *Moving Wheels*, some of these services provide personalized routes. The main limitation of these systems is that they cover relatively small regions; for example *Route4U*[7] can only provide navigation instructions in some parts of Budapest (Hungary) while *Kimap*[8] only covers a few small towns in Italy. This shows that the main challenge with these applications is the large scale collection of geo-referenced information, and indeed our contribution is aimed at mitigating this problem.

Considering the scientific literature, four main challenges have been addressed in the field of navigation for people with disabilities: (a) to compute the user's position with high precision [11]–[14], (b) to compute personalized navigation instructions [2], [3], (c) to effectively convey them (e.g., to blind users) [15], [16], and (d) to detect urban features. This last challenge has been addressed with two different approaches: crowdsourcing and automatic detection techniques. With crowdsourcing, information is manually annotated by

[6]www.mapillary.com
[7]route4u.org
[8]www.kimap.it



(a) Position (FL=front-left, RR=rear-right, RC=rear-center).

(b) Type (A=accelerometer, G=Gyroscope, M=Magnetometer).

(c) Classifiers (RF=flat random forest, HRF=hierarchical random forest, MNB=multinomial naive bayes, MLP=multi-layer perceptron, SVM=support vector machine).

Fig. 10. Performance varying position and type of inertial sensor as well as the classifier. The charts show the average F1 score for coarse-grained labels.

end users or other stakeholders [17], [18] as in the *sidewalk* project[9]. A well-known problem with crowdsourcing is to motivate users to contribute since it often requires an explicit user action. This problem is addressed, among others, by Liu et al. [19] while designing the *WeMap* system [20] that, similarly to *Moving Wheels*, is aimed at providing accessibility information about routes and places. Unfortunately, based on our study, wheelchair users are rarely willing to manually insert accessibility data. As a consequence, only a small fraction of the necessary information is provided, it is often unreliable, and it easily becomes obsolete. Consequently, these services are rarely useful, according the users we interviewed.

Automatic detection of urban features can be adopted to overcome the limitations of crowdsourcing. Computer vision techniques have been shown to be effective to detect some urban features, like pedestrian crossings and traffic lights, both from images captured by the device camera [21], [22], and from satellite images [23]. However, there are other information, like the inclination of a curb ramp, that are harder to detect with this approach and hence we believe that computer vision techniques are complementary to the solution proposed in this paper. An alternative approach to automatically detect urban features is to process inertial data and, to the best of our knowledge, the only solution proposed in the literature is based on data collected from people walking in the urban environment [24], while *Moving Wheels* uses data from wheelchair users.

The machine learning methods we propose and adapt to our application are well known in human activity recognition and have been extensively studied in the literature. Supervised or semi-supervised classification techniques are usually adopted to address this problem [7]. Several works proposed to recognize human activities (walking, running, etc.) by analyzing data from inertial sensors found in commonly available mobile devices, like smartphones, smartwatches or wristbands [25]–[27]. However, activity recognition for wheelchair users is an application domain with its own peculiarities that has been only partially investigated. Smart cushions have been proposed to monitor life-style revealing activities for sedentary subjects (including wheelchair users) [28]. Inertial sensors have also been used to detect simple activities to improve GPS-based localization for both pedestrian and wheelchair users [14]. Differently from those approaches, we rely on inertial sensors to detect activities which in turn disclose detailed information about urban features.

[9]sidewalk.umiacs.umd.edu

VIII. Conclusion

We presented *Moving Wheels*, a urban navigation system for wheelchair users and we proposed a technical solution for automatic crowdsourcing of inertial sensor data enabling the inference of geo-referenced potential obstacles. Our experiments on real data show that the proposed approach is indeed effective.

Future work will be mainly devoted to scale-up the experiments to a larger set of urban features and to a larger training set, including routes in the actual city instead of the real but protected environment used for this work. We also intend to develop a mobile app implementing the proposed technique so that classification can be run in real time, which in turn will allows us to conduct experiments with active learning techniques.

Acknowledgment

We acknowledge Alessandro Fellegara of *Tribe Communication* for inspiring this work.

References

[1] R. Velho, “Transport accessibility for wheelchair users: A qualitative analysis of inclusion and health,” *International Journal of Transportation Science and Technology*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S204604301730117X

[2] L. Beale, K. Field, D. Briggs, P. Picton, and H. Matthews, “Mapping for wheelchair users: Route navigation in urban spaces,” *The Cartographic Journal*, vol. 43, no. 1, pp. 68–81, 2006.

[3] T. Völkel and G. Weber, “Routecheckr: personalized multicriteria routing for mobility impaired pedestrians,” in *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 2008, pp. 185–192.

[4] S. Mascetti, D. Ahmetovic, A. Gerino, and C. Bernareggi, “Zebrarecognizer: Pedestrian crossing recognition for people with visual impairment or blindness,” *Pattern Recognition*, vol. 60, pp. 405–419, 2016. [Online]. Available: https://doi.org/10.1016/j.patcog.2016.05.002

[5] J. Leaman and H. M. La, “A comprehensive review of smart wheelchairs: past, present, and future,” *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 486–499, 2017.

[6] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, p. 33, 2014.

[7] O. D. Lara, M. A. Labrador *et al.*, "A survey on human activity recognition using wearable sensors." *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[8] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 31–72, 2011.

[9] M. Kipp, "Anvil-a generic annotation tool for multimodal dialogue," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[10] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "Smote-rsb*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory," *Knowledge and information systems*, vol. 33, no. 2, pp. 245–265, 2012.

[11] J. Rajamäki, P. Viinikainen, J. Tuomisto, T. Sederholm, and M. Säämänen, "Laureapop indoor navigation service for the visually impaired in a wlan environment," in *Proceedings of the 6th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications*. World Scientific and Engineering Academy and Society (WSEAS), 2007.

[12] M. Nakajima and S. Haruyama, "Indoor navigation system for visually impaired people using visible light communication and compensated geomagnetic sensing," in *Communications in China*. IEEE, 2012.

[13] M. Murata, D. Ahmetovic, D. Sato, H. Takagi, K. M. Kitani, and C. Asakawa, "Smartphone-based indoor localization for blind navigation across building complexes," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018.

[14] M. Ren and H. A. Karimi, "Movement pattern recognition assisted map matching for pedestrian/wheelchair navigation," *The journal of navigation*, vol. 65, no. 4, pp. 617–633, 2012.

[15] D. Ahmetovic, U. Oh, S. Mascetti, and C. Asakawa, "Turn right: Analysis of rotation errors in turn-by-turn navigation for individuals with visual impairments," in *Conf. on Computers and Accessibility*. ACM, 2018.

[16] G. Dubus and R. Bresin, "A systematic review of mapping strategies for the sonification of physical quantities," *PLoS ONE*, vol. 8, no. 12, p. e82491, Dec. 2013.

[17] K. Hara, S. Azenkot, M. Campbell, C. L. Bennett, V. Le, S. Pannella, R. Moore, K. Minckler, R. H. Ng, and J. E. Froehlich, "Improving public transit accessibility for blind riders by crowdsourcing bus stop landmark locations with google street view: An extended analysis," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 6, no. 2, p. 5, 2015.

[18] M. Saha, K. Hara, S. Behnezhad, A. Li, M. Saugstad, H. Maddali, S. Chen, and J. E. Froehlich, "A pilot deployment of an online tool for large-scale virtual auditing of urban accessibility," in *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 2017, pp. 305–306.

[19] Z. Liu, S. Shabani, N. G. Balet, M. Sokhn, and F. Cretton, "How to motivate participation and improve quality of crowdsourcing when building accessibility maps," in *Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual*. IEEE, 2018, pp. 1–6.

[20] Z. Liu, N. G. Balet, M. Sokhn, and E. De Gaspari, "Crowdsourcing-based mobile application for wheelchair accessibility," *Journal on Technology & Persons with Disabilities*, 2017.

[21] S. Mascetti, D. Ahmetovic, A. Gerino, and C. Bernareggi, "Zebrarecognizer: Pedestrian crossing recognition for people with visual impairment or blindness," *Pattern Recognition*, 2016.

[22] S. Mascetti, D. Ahmetovic, A. Gerino, C. Bernareggi, M. Busso, and A. Rizzi, "Robust traffic lights detection on mobile devices for pedestrians with visual impairment," *Computer Vision and Image Understanding*, 2016.

[23] D. Ahmetovic, R. Manduchi, J. M. Coughlan, and S. Mascetti, "Mind your crossings: Mining gis imagery for crosswalk localization," *ACM Transactions on Accessible Computing (TACCESS)*, 2017.

[24] A. Bujari, B. Licar, and C. E. Palazzi, "Movement pattern recognition through smartphone's accelerometer," in *Consumer communications and networking conference (CCNC), 2012 IEEE*. IEEE, 2012, pp. 502–506.

[25] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.

[26] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber, "Smartwatch-based activity recognition: A machine learning approach," in *Biomedical and Health Informatics (BHI), 2016 IEEE-EMBS International Conference on*. IEEE, 2016, pp. 426–429.

[27] T. Brezmes, J.-L. Gorricho, and J. Cotrina, "Activity recognition from accelerometer data on a mobile phone," in *International Work-Conference on Artificial Neural Networks*. Springer, 2009, pp. 796–799.

[28] C. Ma, W. Li, R. Gravina, J. Cao, Q. Li, and G. Fortino, "Activity level assessment using a smart cushion for people with a sedentary lifestyle," *Sensors*, vol. 17, no. 10, p. 2269, 2017.