Real-Time Fall Detection and Activity Recognition Using Low-Cost Wearable Sensors

Cuong Pham[1,2] and Tu Minh Phuong[1]

[1] Department of Computer Science,
Posts & Telecommunications Institute of Technology,
Ha Noi, Vietnam
cuongpham.ptit@gmail.com, phuongtm@ptit.edu.vn
[2] Affiliation: Culture Lab, Newcastle University, Newcastle upon Tyne, NE17RU, UK

Abstract. We present a real-time fall detection and activity recognition system (FDAR) that can be easily deployed using Wii Remotes worn on human body. Features extracted from continuous accelerometer data streams are used for training pattern recognition models, then the models are used for detecting falls and recognizing 14 fine grained activities including unknown activities in realtime. An experiment on 12 subjects was conducted to rigorously evaluate the system performance. With the recognition rates as high as 91% precision and recall for 10-fold cross validation and as high as 82% precision and recall for leave one subject out evaluations, the results demonstrated that the development of real-time fall detection and activity recognition systems using low-cost sensors is feasible.

1 Introduction

Falls are one of most high-risk problems for old people. A study conducted by the Centers for Disease Control and Prevention [1] shows that up to 33% adult people aged over 65 falls at least once a year. Of which about 30% cases cause medium to severe injuries that can lead to death. This is an obstacle on the elderly's living independent at homes. Although there are a number of previous studies on fall detection and yielded significant results (i.e. accuracy of 80-90%), in fact falls might occur while the elderly performs daily life activities such as walking, jumping, going-up stair, going-down stair, running etc. The information of such activities can provide warning the elderly for preventing the falls (i.e. jumping might highly cause a fall). Moreover, a study low-level daily activity information can not only be a fundamental element for situated services to support people [2, 23], but also be useful for health & energy expenditure monitoring [3]. The development of system integrated both activity recognition and fall detection with low-cost technologies at people's homes has real potential to provide age-related impaired people with a more autonomous lifestyle, while at the same time reducing the financial burden on the state and these people and their families.

Majority of existing activity recognition systems and fall detection systems require specific hardware and software design which often cost hundreds of US dollars or

B. Murgante et al. (Eds.): ICCSA 2013, Part I, LNCS 7971, pp. 673–682, 2013.

more. Moreover, pervasive sensing deployment often requires many sensors in the environmental surroundings [3], which might exceed the budgets of the poor and middle-class people. This is especially true in Vietnam where up to 29% of Vietnamese population is classified as poor or below (according to the UNDP standard [4]). Therefore proposing a low-cost sensing and easy deployment technology for prototyping the FDAR system is a key component of this study to make the pervasive computing technologies support people's lives and help the elderly to live more independent at their homes.

Our contribution is twofold:

First, we prototype a fall detection and activity recognition system that can automatically detect human falls and can recognize 13 activities in real-time. Our work is distinct from other works on activity recognitions [7][11][14][25] as we address a set of low-level activities (rather than high-level activity set) and utilize the use of easily deployable and low-cost wearable sensors.

Second, we evaluate the system on an open-dataset (i.e. including unknown activities) collected from 12 subjects who performed 12 activities and 144 falls at their homes under 10-fold cross validation and leave one-subject out protocols.

2 Related Work

Activity recognition

Two common approaches to activity recognition are computer vision based and sensors based. In the first approach, computer vision technology is used to analyze the video streams from digital cameras installed in the environments (i.e. [11][12]) to infer human activities. In the second approach, activity recognition is performed by analyzing the sensing data from sensor streams. Ubiquitous sensors can be embedded into objects [2][13], environments [14][15], or worn on different parts of human body [3][5][6][7]. In this study, we focus on the wearable sensing as embedding sensors into objects or environments often requires many sensors while cameras invade people's privacy. Among sensors available on the market we choose Wii Remotes as they are cheap and easy deployment. Previously, Wii Remotes were also used for recognizing food preparation activities [13].

Fall detection

Majority of fall detection applications are implemented on smart-phone [16][17] which utilizes the acceleration data from accelerometer integrated inside the phone. Although those studies have significant results (~90% accuracy), it is noticed that falls often occur while people are doing other activities such as jumping, running, going-up stairs, going-down-stair etc. Some of them might lead to the elderly fall. Previously, a small number of research addressed fall detection and activity recognition problem (i.e. [18]), but using cameras that can invade people's privacy and without real-time implementation. Moreover, computer vision approach's performance is significant affected by light conditions in the environment (i.e. how it can work at night?). To our knowledge, no existing studies explored the recognition a set of low-level activities and detection of falls using low-cost sensing technologies with real-time implementation.

3 Hardware

In contrast to most of accelerometers often used in research labs (i.e. [4] [5] [6][7]) or on the market but quite expensive (i.e. [8][9][10]), or relatively complex deployment (i.e. requires base-station for communication to the computer), Wii Remotes are relatively cheap, available on the market, and simple deployment as Wii communicates with the computer via a Bluetooth dongle (also very cheap).

The Wii Remote [19] is a consumer off-the-shelf wireless sensing system and games controller which supports two functionalities of relevance to our application: (i) input detection through an embedded accelerometer; and (ii) data communications through Bluetooth. A Wii Remote comprises a printed circuit board (which is encapsulated by a white case) and uses an AXDL 330 accelerometer [3] and a Broadcom BCM2042 chip that integrates the entire profile, application, and Bluetooth protocol stack. Based on Micro Electro Mechanical System (MEMS) technology the AXDL 330 accelerometer is a small, low power, 3-axis accelerometer with signal conditioned voltage outputs. The AXDL 330 accelerometer can sense acceleration in three axes with a minimum full-scale range of ±3g. While the static acceleration of gravity can be used to implement tilt-sensing in applications, dynamic acceleration measurement can be detected through the quantifiers of motion, shock or vibration.

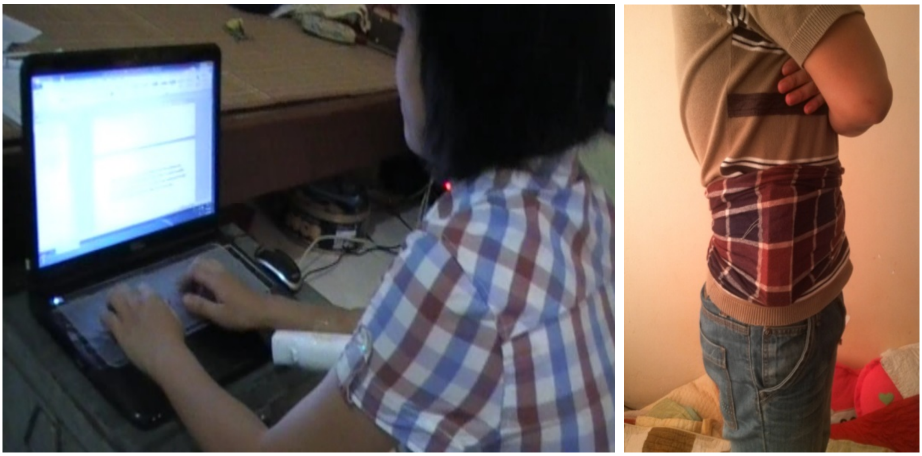

Fig. 1. Wii Remote worn on wrist (left) and Wii Remote worn on hip (right)

The Broadcom BCM2042 board is a system-on-chip which integrates an on-board 8051 microprocessor, random access memory/read only memory, human interface device profile (HID), application, and Bluetooth protocol stack. Furthermore, multiple peripherals and an expansion port for external add-ons are embedded on the board. The integration of these components and the technology's adoption in a mass market consumer games console has significantly reduced the cost of BCM2042. The Wii

Remote's input capabilities include buttons, an infrared sensor and an accelerometer. The infrared sensor is embedded in a camera which detects IR light coming from an external sensor bar. The accelerations are measured in X, Y, and Z axes (relative to the accelerometer) and the three directions of the movement (X, Y, Z) can be computed through tilt angles. Wii Remote inputs and sensor values are communicated to a Bluetooth host through the standard Bluetooth HID protocol. Values for acceleration are transmitted with a sampling frequency of 100Hz (100 samples per second).

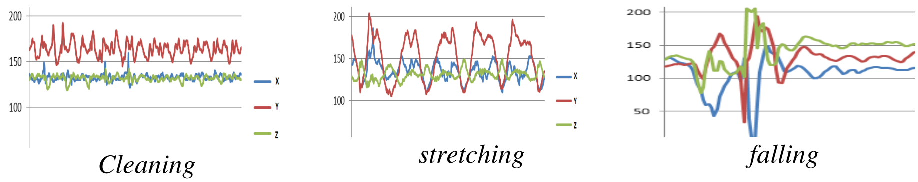

Fig. 2. Examples of accelerometer signals for three human activities and falls

In this study, subjects were asked to worn 2 Wii Remotes: one worn on hip and the other worn on right-hand's wrist. While the sensor worn on hip can provide good features for the detection of falls, running, walking, going-up stairs, the sensor worn on wrist might be useful for recognizing activities performed with hand such as cleaning, typing, and brushing. We will use the combination of both sensing data streams from 2 sensors for the detection of falls and recognition of activities.

4 Real-Time Fall Detection and Activity Recognition

The real-time FDAR algorithm works in 4 steps:

- Signal processing: sensing data is filtered for removing the noise or resampling the lost data.
- Segmentation: a sliding window/frame is used to segment the signal stream into frames of fixed length.
- Feature extraction: from each frame, different features are extracted.
- Classification: the system uses the features extracted from the previous step as input for a HMM based classifier.
- In the following sections, we describe each of these steps in detail.

4.1 Signal Processing

Sensing data from sensors are often noisy and ambiguity. Ideally, at a sampling frequency of 100Hz each second contains 100 samples of X, Y, Z acceleration triplets (i.e. one sample per 10 miliseconds). In practice, real-world factors mean that some samples are lost or dropped (e.g. metallic items placed between the sensors and the receiver). Furthermore, the sensors themselves can yield noisy readings (e.g. too large or small). In such cases, a filter is applied to remove noise and to fill out lost samples. In this step, the data filter performs both a low-pass filtering (removing abnormally low sample values) and a high-pass filtering (removing abnormally high sample values). After that, samples are grouped into sliding windows or frames. If a frame contains less that 75% of its full complement, it is discarded on the grounds that there is insufficient information to classify activities. Otherwise, it is resampled using a cubic spline interpolation method [20] to fill out the lost samples.

Along with acceleration X,Y,Z, we compute pitch, roll for each triplet:

$$Pitch = 2\ arctan\left(\frac{y}{\sqrt{x^2+z^2}}\right) \quad (1)$$

$$Roll = 2\ arctan\left(\frac{x}{\sqrt{y^2+z^2}}\right) \quad (2)$$

where x, y, z are acceleration values of the three axis

4.2 Segmentation

Previous studies showed that the length of sliding window has significantly impact on the performance of the pattern recognition algorithms [5][13]. In this study, we did a pilot study on the subset of collected dataset for selecting a reasonable length for sliding window. We varied the window length 1 second, 1.2 second, 1.5 seconds, 1.8 seconds, 2 seconds and 2.5 seconds and we stick on the window length of 1.8 seconds. The reason for the choosing window length of 1.8 second is that this length allows avoiding delay from continuously real-time processing while providing a reasonable recognition rate.

4.3 Feature Extraction

For each frame of size*n* where n is number of time points, the following features are extracted:

$$Mean(x) = \frac{\sum_{i=1}^{n} x_i}{n} \quad (3)$$

$$Standard\ deviation(x){:}\ \delta_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i^2) - [Mean(x)]^2} \quad (4)$$

$$Energy(x) = \frac{\sum_{i=1}^{n} x_i^2}{n} \quad (5)$$

$$Entropy(x) = -\sum_{i=1}^{n} p(x_i) log\ (p(x_i)) \quad (6)$$

where x_i is an acceleration value, $p(x_i)$, a probability distribution of x_i within the sliding window, can be estimated as the number of x_i in the window divided by n.

$$Correlation(X,Y) = \frac{\mathrm{cov}(x,y)}{\delta_x \delta_y} \quad (7)$$

In which cov(x,y) is covariance and δ_x , δ_y are standard deviations of x and y.

Peak/bottom acceleration: for each sliding window, we also extracted 3 peak values and 3 bottom values of acceleration.

These features are combined into a 58-dimentional feature vector, composed of *Mean X, Standard deviation X, Energy X, Entropy X, Mean Y, Standard deviation Y, Energy Y, Entropy Y, Mean Z, Standard deviation Z, Energy Z, Entropy Z, Mean Pitch, Standard deviation Pitch, Energy Pitch, Entropy Pitch, Mean Roll, Standard deviation Roll, Energy Roll, Entropy Roll, Correlation XY, Correlation YZ, Correlation ZX, peak value of X, peak of Y, peak of Z, bottom of X, bottom of Y*, and *bottom value of Z*. These feature vectors are then used for training the pattern recognition algorithms. In the next section, we present the hidden Markov models [24] that we employed for real-time fall detection and activity recognition.

4.4 Hidden Markov Model-Based Classifier

In brief, a hidden Markov model [24] is a stochastic model that can be used to characterize statistical properties of a sensing signal. An HMM is associated with a stochastic process not directly observable (i.e. hidden), but it can be observed indirectly through a set of other outputs. The key problem is to determine the hidden parameters given a model and observedparameters. Then, the extracted model parameters can be used to perform further analysis in future learning. An HMM is based on the Markov assumption that the current state depends only on the precedingstate.

In our domain, we use HMMs (one HMM for one activity) with a mixture of Gaussians of for state's observation distribution. The number of hidden statesis manually tailored for each model. In the training phase, the parameters of each model (i.e. initial state probabilities, state transition probabilities, observation probability distribution) were estimated using the Baum-Welch algorithm (implemented in Murphy's HMM ToolKit [25]). After that, we use the trained models for classifying the falls and activities. The Viterbi algorithm is implementedand is used for the computation of the log likelihood probability of each observation sequence O (i.e. a feature vector computed from a continuous sliding window) given the trained model. The classifier will choose the model that produces the maximum log likelihood given feature vectors computed from test data.

5 Experimental Evaluation

5.1 Data Collection and Annotation

Twelve students from our institute (Post &Telecommunication Institute of Technology) were recruited, each wore 2 Wii Remotes, one on wrist and the other on hip. Each student was asked to perform 12 activities including walking, jumping, going-up stair, going-down stair, running, stretching, cleaning, typing, standing-to-sit, sitting-to-stand, brushing teeth, vacuuming and 12 (intentional) falls with various postures. No order of performing activities is required and no time constraint to each activity performed by the subject. Each activity is required to be performed as naturally as possible. In addition to sensing data, we recorded videos of subjects performing the activities.

To synchronize between the collected videos and the accelerometer data from Wii remotes, at the beginning time of each session the subject was asked to shake the body and the hand 3 times to make distinct signals. The subject was apparently shown on the videos. In addition, along with each sample, a timestamp was written to the acceleration data log files.

The subjects were given a list of 12 labels of activities to annotate the collected videos using ELAN Multimedia Annotator Tool [22]. Movements that are not one of 12 activities and falls were automatically labeled with “unknown”.

5.2 Performance Metrics

Recognition results are reported as frame-wise precision, recall and F-measure values. The *precision* for an activity was calculated by dividing the number of correctly classified frames by the total number of frames classified as being a particular activity (i.e. *true positives*/(*true positives* + *false positives*)). *Recall* was calculated accordingly as the ratio of the number of correctly classified frames to the total number of frames of an activity (i.e. *true positives*/*total number of frames of an activity*). And, F-measure is the harmonic mean of precision and recall.

5.3 Results for 10-Fold Cross Validation

Under 10-fold cross validation procedure, the dataset was randomly partitioned into 10 parts of equal size. Nineof them are used for training and the remaining one is used for testing. Then, the process is repeated for all 10 parts and the results are averaged. The results are shown on the Table 1. Note that both training and test sets may contain data from the same subject.

Table 1. -fold cross validationresults (numbers are in percent)

Activity	Precision	Recall	F-measure
brushing teeth	94.64	90.56	92.56
cleaning	89.98	82.43	86.04
falling	93.06	91.67	92.36
going-down stairs	96.76	93.06	94.87
going-up stairs	98.72	95.3	96.98
jumping	98.48	97.98	98.23
running	97.39	96.52	96.95
sitting-to-stand	87.43	89.22	88.32
standing-to-sit	96.75	94.16	95.44
stretching	76.71	69.41	72.88
typing	96.4	95.89	96.14
vacuuming	97.92	97.51	97.71
walking	96.77	95.35	96.05
unknown	86.17	86.63	86.4
Average	**92.58**	**90.54**	**91.55**

Overall, precision, recall, and F-measure are over 90% for 10-fold cross validation(i.e. subject dependent) analysis. Majority number of activities including falls has precision and recall as high as over 90% except for stretching activity which is often misclassified as cleaning. It is noticed that the recognition rate of falling is 93% precision and 91.6% recall.

5.4 Results for Leave-One-Subject-Out Evaluation

In addition to 10-fold cross validation which is often used for systems for personal use or adaptation. We envisage to evaluate the system under the *leave-one-subject-out* protocol. In which, we used 11 subjects for training and left the remaining one for testing. The process was repeated for all 12 subjects, and the results were averaged. Table 2 shows the results. It is noticed that the tested subject was not included in the training data.

Table 2. Leave-one-subject-out results (numbers are in percent)

Activity	Precision	Recall	F-measure
brushing teeth	85.71	86.99	86.35
cleaning	81.2	77.5	79.31
falling	84.03	82.64	83.33
going-down stairs	91.67	84.72	88.06
going-up stairs	94.44	90.17	92.26
jumping	97.47	96.46	96.96
running	96.23	93.91	95.06
sitting-to-stand	73.65	74.85	74.25
standing-to-sit	78.57	77.92	78.24
stretching	64.38	63.93	64.15
typing	95.37	95.12	95.24
vacuuming	96.05	85.45	90.44
walking	95.15	88.89	91.91
unknown	72.25	70.51	71.37
Average	**85.2**	**82.16**	**83.65**

The overall recognition rates for leave one-subject out evaluation are 85% precision, 82% recall, and 83.6% F-measure, which are lower than thoseof 10-fold cross-validation. Note that, leave-one-subject-out is more difficult then cross-validation because the system has to recognize activities for unseen subjects. This setting is also more similar to practical conditions where a system trained on a set of subjects is used to make recognitions for new subjects, data about which are unknown at training time. Again,stretching proved to be the most difficult activity to recognize with the F-measure as low as 63% while for running, jumping, and typing the system achieved F-measures higher than 90%.This is consistent with 10-fold cross validation results.

6 Conclusion

We have presented a solution for detecting and recognizing falls and 12 other human activities. Our method uses inexpensive sensing devices such as Wii Remotes worn on human body as sources of signals, thus provides a low-cost solution. From accelerometer signals, the system extracts several types of features that summarize different aspects of movements. A hidden Markov model is used to map these features into hidden states, which corresponds to different activities. An empirical study with data collected from 12 subjects demonstrates the effectiveness of the proposed method. The system achieved average F-measures of 91.55% for 10-fold cross-validation and 83.6% for leave-one-subject-out settings respectively. With relatively high accuracy while being simple and inexpensive, the proposed solution can be used for practical applications requiring the recognition of human activities.

References

1. Falls Among Older Adults: An Overview, `http://www.cdc.gov/HomeandRecreationalSafety/Falls/adultfalls.html` (accessed on January 28, 2013)
2. Pham, C., Hooper, C., Lindsay, S., Jackson, D., Shearer, J., Wagner, J., Ladha, C., Ladha, K., Plötz, T., Olivier, P.: The Ambient Kitchen: A Pervasive Sensing Environment for Situated Services. In: Demonstration in the Designing Interactive Systems Conference (DIS 2012). ACM, Newcastle Upon Tyne (2012)
3. Albinali, F., Intille, S., Haskell, W., Rosenberger, M.: Using Wearable Activity Type Detection to Improve Physical Activity Energy Expenditure Estimation. In: Proceedings of the 12th ACM International Conference on Ubiquitous Computing (Ubicomp 2010), pp. 311–320. ACM, New York (2010)
4. General office for population family planning, `http://www.gopfp.gov.vn` (accessed on January 28, 2013)
5. Bao, L., Intille, S.: Activity Recognition from User-Annotated Acceleration Data. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004)
6. Ravi, N., Dandekar, N., Mysore, P., Littman, M.L.: Activity Recognition from Accelerometer Data. In: Porter, B. (ed.) Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence (IAAI 2005), vol. 3, pp. 1541–1546. AAAI Press (2005)
7. Huỳnh, T., Blanke, U., Schiele, B.: Scalable Recognition of Daily Activities with Wearable Sensors. In: Hightower, J., Schiele, B., Strang, T. (eds.) LoCA 2007. LNCS, vol. 4718, pp. 50–67. Springer, Heidelberg (2007)
8. SUNSPOT, `http://www.sunspotworld.com/products/` (accessed on January 28, 2013)
9. WAX3, `http://axivity.com/v2/` (accessed on January 28, 2013)
10. Microstrain, `http://www.microstrain.com/parameters/acceleration` (accessed on January 28, 2013)
11. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., James, M.R.: A Scalable Approach to Activity Recognition based on Object Use. In: Proceedings of the 11th International Conference on Computer Vision (ICCV 2007), pp. 1–8. IEEE Press, Rio de Janeiro (2007)

12. Duong, T.V., Bui, H.H., Dinh, Q.P., Venkatesh, S.: Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 838–845. IEEE Computer Society, Washington, DC (2005)
13. Pham, C., Olivier, P.: Slice&Dice: Recognizing Food Preparation Activities using Embedded Accelerometers. In: Proc. Europ. Conf. Ambient Intell., pp. 34–43
14. Buettner, M., Prasad, R., Philipose, M., Wetherall, D.: Recognizing Daily Activities with RFID-based Sensors. In: 11th International Conference on Ubiquitous Computing (Ubicomp 2009), pp. 51–60. ACM, New York (2009)
15. Tapia, E.M., Intille, S.S., Larson, K.: Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 158–175. Springer, Heidelberg (2004)
16. Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P.V.: A Smartphone-based Fall Detection System. Pervasive Mob. Comput. 8(6), 883–899 (2012)
17. Sposaro, F., Tyson, G.: iFall: An Android Application for Fall Monitoring and Response.In. In: Proceedings of 31st Annual International Conference of the IEEE EMBS, Minneapolis, Minnesota, USA (2009)
18. Olivieri, D.N., Conde, I.G., Sobrino, X.A.V.: Eigenspace-based Fall Detection and Activity Recognition from Motion Templates and Machine Learning. Expert Syst. Appl. 39(5), 5935–5945 (2012)
19. Wii Remote, `http://en.wikipedia.org/wiki/Wii_Remote` (accessed on January 28, 2013)
20. Spline Interpolation, `http://en.wikipedia.org/wiki/Spline_interpolation` (accessed on January 29, 2013)
21. Ward, A.J., Lukowicz, P., Gellersen, W.H.: Performance Metrics for Activity Recognition. ACM Trans. Intell. Syst. Technol. 2(1), Article 6, 23 pages (2011)
22. ELAN, `http://tla.mpi.nl/tools/tla-tools/elan/` (accessed on February 19, 2013)
23. Wherton, P.J., Monk, F.A.: Technological Opportunities for Supporting People with Dementia Who Are Living at Home. Int. J. Hum.-Comput. Stud. 66(8), 571–586 (2008)
24. Rabiner, L.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In: Waibel, A., Lee, K.-F. (eds.) Readings in Speech Recognition, pp. 267–296. Morgan Kaufmann Publishers Inc., San Francisco
25. Kasteren, T., Noulas, V., Englebienne, A., Kröse, G.: Accurate Activity Recognition in a Home Setting. In: Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp 2008), pp. 1–9. ACM, New York (2008)
26. HMM ToolKit, `http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html` (accessed on March 07, 2013)