An Embedded Ground Change Detector for a "Smart Walker"

Weiss Velandia, Viviana Lucia; Korolev, Aleksandr; Bologna, Guido; Cloix, Séverine; Pun, Thierry

How to cite

WEISS VELANDIA, Viviana Lucia et al. An Embedded Ground Change Detector for a 'Smart Walker'. In: Artificial Computation in Biology and Medicine. Elche, Spain. [s.l.] : Springer, 2015. p. 533–542. (Lecture Notes in Computer Science) doi: 10.1007/978-3-319-18914-7_56

This publication URL: https://archive-ouverte.unige.ch//unige:74463
Publication DOI: 10.1007/978-3-319-18914-7_56

An Embedded Ground Change Detector for a "Smart Walker"

Viviana Weiss[1], Aleksandr Korolev[1], Guido Bologna[1], Séverine Cloix[1,2], Thierry Pun[1]

[1] Computer Science Department, University of Geneva, Geneva, Switzerland
{viviana.weiss,Aleksandr.Korolev,guido.bologna,thierry.pun}@unige.ch
[2] Centre Suisse d'Electronique de de Microtechnique, CSEM, Neuchâtel, Switzerland
severine.cloix@csem.ch

Abstract. Millions of elderly people around the world use the walker for their mobility; nevertheless, these devices may lead to an accident. One of the cause of these accidents is misjudge the terrain. The main objective of this work is the implementation of a ground change detector in real time on a small and light embedded system that can be clipped on a rollator. As a long-term goal, this device will allow users to anticipate entering dangerous situations. We implemented an algorithm to detect ground changes based on color histograms and texture descriptor given as inputs to multi-layer perceptrons. Experiments were performed both off-line and with an embedded system. The obtained results indicated that it is possible to have an accurate detector which is able to distinguish ground changes in real-time.

Keywords: Ground change detector, Embedded system, Artificial Neural Network (ANN), Elderly care, Gerontechnology.

1 Introduction

The world is facing a situation without precedents. The proportion of old people and the expectation of life increase everywhere. The number of elederly people is projected to grow up to more than 2 billions, by 2050 [1]. Reducing severe disability from disease and health conditions is one key to holding down health and social costs.

The rollator (Fig. 1), widely spread among elderly, aims at helping users keep their independence and mobility. However, these tools can lead to falls, especially in urban areas and buildings. They occur when the user misjudges the nature of ground, which can happen in any kind of familiar or unknown environments. Approximately, 87% of elderly people falls are attributable to walkers use [2].

Various prototypes of "intelligent walkers" are motorized, equipped with route planning and obstacle detection, relying on active sensing (laser, sonar, IR ligth), or passive sensing (RFID tags, visual signs) [3–5]. Such aids are complex, thus expensive and exist only at prototype level. In practice, their use is limited to indoor situations due to their short battery life.

Fig. 1: Typical walker with four wheels, handles with breaks and a seat.

In this work, we present the *"EyeWalker"* project which aims at developing a low-cost, ultra-light computer vision-based prototype for users with mobility problems. This device is meant to be small and lightly embedded; it would be easily fixed on a standard rollator. One of the goals is to warn users before they enter in a dangerous ground. It has to operate in indoor and outdoor environments. The users initially targeted by this project are elderly persons that still live independently.

In this work, our goal is to asses the accuracy of our ground change detector and to determine at which image resolution it could be used in a real time implementation. Our key idea is based on the estimation of changes of brightness, color and textures under real environmental conditions and on the reduction of image resolution, in order to reduce the time processing.

This paper is organized as follows: Section 2 describes the material and the methodology to detect ground changes. Section 3 presents the experiments involving off-line and embedded systems at different resolutions, before concluding remarks.

2 System Design

This section briefly describes the methodology to detect the ground change, the different hardware set-up used for the off-line and embedded systems and the datasets used to compare both systems.

2.1 Detection process

We would like to prevent in real time the falls related to the loss of balance caused by the ground change. As a result, the ground change detection is based on the comparison between the current frame and the average of k previous frames.

The procedure is divided in two different steps. Firstly, we extract the image descriptor from the current frame and the average of the k previous frames. Secondly, we use an Artificial Neural Network (ANN) trained on color histograms

and a texture descriptor to decide whether to warn the user. The block diagram to detect ground changes is illustrated in Fig.2.

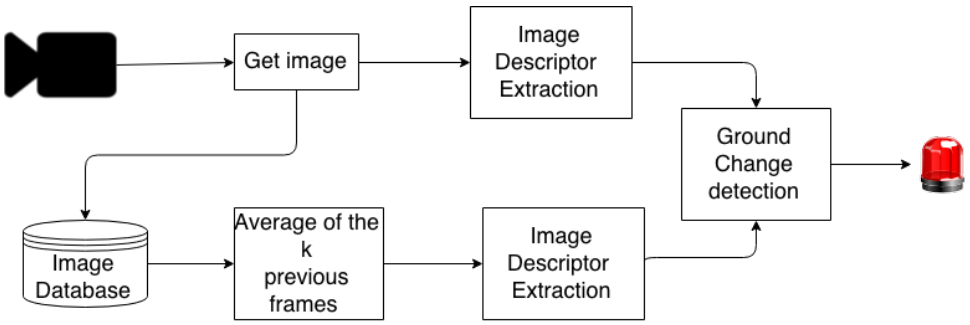


Fig. 2: General block diagram to detect ground changes.

Image Descriptor

From each video frame, we calculate an image descriptor based on colors and textures. Specifically, a similarity measure between the current image and the k previous image is determined and provided to a neural network.

For the color feature, different types of color space were tested in [6]. As a result, the HSV color space demostrated to be the most suitable. From the image, we obtain the normalized color histogram h_c for each color channel using the following equation:

$$h_{c_i}^{H,S,V} = \frac{n_i}{N}, i = 0, \ldots, 255 \tag{1}$$

where n_i is the number of pixels with color label i and N is the total number of pixels in the image for each channel.

As a texture feature, the Local Edge Pattern (LEP) was used. LEP describes the spatial structure of the local texture according to the organization of edge pixels. To compute the LEP histogram, an edge image must be obtained first. The edge image is obtained by applying the Sobel edge detector to intensity gray level. The binary values are then multiplied by the corresponding binomial weights in a LEP mask, and the resulting values are summed to obtain the LEP value.

The LEP value is defined as [7],

$$LEP(n, m) = \sum_{i,j \in I} K_e(i, j) \times I_e(n, m) \tag{2}$$

where $I_e(n, m)$ denotes the binary image, K_e is the LEP mask and $LEP(n, m)$ is the LEP value for the pixel (n, m). The LEP mask is given by:

$$K_e = \begin{pmatrix} 1 & 2 & 4 \\ 128 & 256 & 8 \\ 64 & 32 & 16 \end{pmatrix} \tag{3}$$

Finally, the LEP normalized histogram h_e can be computed from

$$h_{e_i} = \frac{n_i}{N}, i = 0, \ldots, 511 \tag{4}$$

where n_i is the number of pixels with LEP value i and N is the total number of pixels in the image.

Fig.3 shows the methodology to extract the Image Descriptor. For each new frame we start from the calculation of the average image of the k previous frames. The next step consists in the processing of the current frame and the averaged image. Specifically:

- we apply the blur filter to reduce image noise and reduce detail;
- we convert the image from the RGB to the HSV color space;
- we calculate the histograms for H, S and V color variables;
- we transform the RGB image into a gray level image;
- we apply the Sobe filter to detect the edges;
- we use the LEP filter and calculate the LEP histogram.

We take into account the values of each bin in the histograms (H,S,V channels and LEP) as an image descriptor. Since we have three color channels, each channel having 256 bins and a texture channel represented by 512 bins, the final Image Descriptor contains 2560 bins (Fig. 4). Principal Component Analysis (PCA) is used to reduce the dimensionality of the image descriptor to 200 bins.

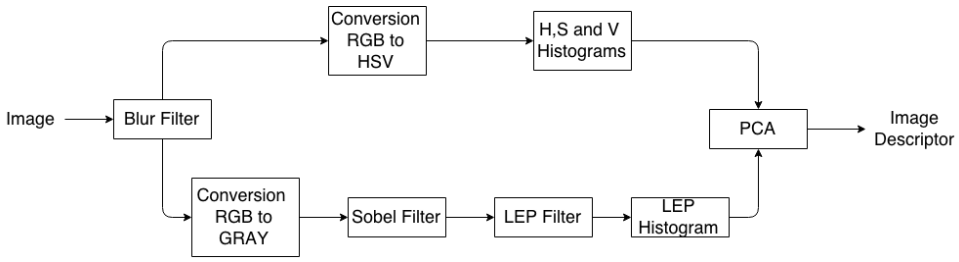


Fig. 3: Block diagram to extract Image Descriptor.

Ground Change Detector

The color's distribution and LEP are used to obtain a distance measure between two images characterising the inhomogeneity of a surface. Specifically, this is calculated between the current image and the average of the k previous images. To measure the similarity between frames, several methods were presented

in [8]. Here, to distinguish the ground change we implemented an Artificial Neural Network (ANN).

We used a multilayer perceptron with n neurons in the input layer, m hidden layers and 2 neurons in the output layer. Values for n and m were determined empirically in [6]. The output layer has two neurons, one represents the detection of the ground change and the other indicates unchanged ground. In this neural network, the input vector theoretically is represented as a vector of 2560 neurons (see Fig. 4). In order to fulfil the real time requirements, the size of the input vectors was reduced to 200 neurons by means of PCA.

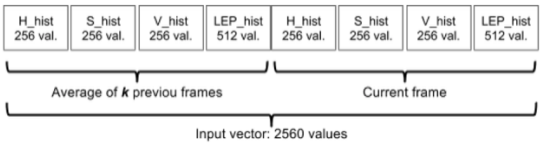


Fig. 4: Image Descriptor configuration.

2.2 Set-up

A set of outdoor video sequences were collected from a campus path at Geneva University using the walker shown in Fig.1. This data set was recorded with two set-ups, the first set-up (video set-up I) is a walker equipped with a color webcam (Logitech HD Webcam C510, 8 Mpixels) with a resolution of 640x480 pixels. The second set-up (video set-up II) is based on the camera sensor "Caspa VL" (Fig.7(a)) with a image resolution of 752x480 pixels. The cameras were located 60 cm from the ground. The covered visual field region is about 130 cm long, as shown in Fig.5. We use a total of six videos recorded at different times for the first set-up, each video containing between 188 and 313 frames and two to four ground change transitions. For the second set-up, 25 videos were recorded with 1687 frames in total. At the Fig.6 shows an example of both video sets. Note that videos were recorded at an approximative speed of 0.6 m/s (2.3 km/h) with a frame rate of $25 fps$.

The final system is composed by the camera sensor and the computer. In the first set-up, the color webcam is connected to a Dell computer with an Intel(R) Core(TM) i7-2600 CPU (3.4 GHz). For the second set-up, the detector was implemented in the Linux operating system with the use of the OpenCV library [9]. Specifically, the camera sensor "Caspa VL" [10] is connected to a "Tobi" platform (Fig.7(b)) [11] with Overo® (Computer On-Module) coard [12]. Tobi and Overo® work with a ARM Cortex-A8 (Texas Instruments DaVinci DM3730 up to 1GHz capable) CPU with 512MB RAM.

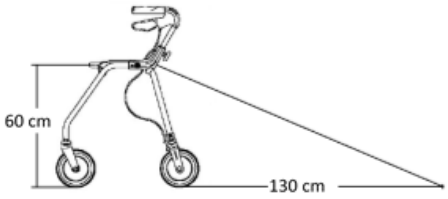


Fig. 5: Walker set-up for the detection of ground changes.

(a) Video set-up I

(b) Video set-up II

Fig. 6: Video sets image examples.

(a) Caspa VL

(b) Tobi Platform

Fig. 7: Hardware embedded system

3 Experiments

3.1 Implementation Comparisons

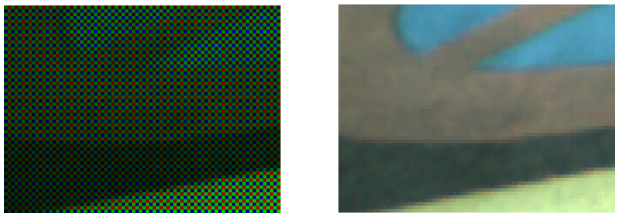
One of the goals of this research was to investigate the plausibility of migrating from the off-line system (Matlab) to embedded system (C++).

During the calculation of the histograms a substantial difference between the results obtained in MatLab and OpenCV was discovered. For the color feature, differences in the HSV conversion between both systems were found. On one hand, the biggest difference is observed in the S component and the smallest is for the V component.

On the other hand, differences in the implementation of the Sobel edge detection operator and gray scale images in MatLab and OpenCV are the main

reasons of the discrepancy in the textural descriptor. One of the first major problems encountered during the migration process was the different implementation of the Sobel Filter. The implementation of the standard method for the Sobel edge detection in MatLab and OpenCV is quite different. In MatLab the result is a binary image, whereas in OpenCv it is a gray level image. Moreover, the standard Sobel function does not give any binary image and requires the additional applying of the threshold and "skeleton" in OpenCV. Because of this, there is a significant difference that is characterized by a Peak Signal to Noise Ratio (PSNR) equal to $64.46dB$. PSNR is an unit to measure the image distortion.

Other source of difference is the sensor camera. The camera Caspa VL is not a camera of full value. The pictures produced by the Caspa sensor are the results of applying the Bayer filter (see example at Fig. 8).



(a) Before processing (b) After processing

Fig. 8: Bayer filter result

The Bayer filter mosaic is a color filter array (CFA) for arranging RGB color filters on a square grid of photo sensors. The raw output of Bayer-filter sensors is referred to as a Bayer pattern image. Since each pixel is filtered to record only one of three colors, the data from each pixel cannot fully specify each of the red, green, and blue values on its own.

3.2 Evaluation

To assess the performance of our approach, an extensive and systematic evaluation in terms of accuracy and processing time of image resolution were conducted on a data set labelled manually.

The two classes, ground change and no change, are defined as follows: a frame is labelled positive as soon as a ground change enters the visual field and it remains positive until the user is completely on the new terrain; the frame is labelled negative, otherwise [6]. To compare the methods, we use the confusion matrix shown in Table 1, where accuracy is defined as:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{5}$$

Table 1: Terminology use for the evaluation

		Condition	
		Ground change	No change
System result	Ground change	True positive "*tp*" (Correct alarm)	False Positive "*fp*" (Unexpected alarm)
	No change	False negative "*fn*" (Missing alarm)	True negative "*tn*" (Correct absence of alarm)

In our system, we have two critical values. The first one is the missing alarm because it can generate an accident. We however must minimize the false positive rate to ensure user acceptance. And the second one is the processing time.

We tested our detector in both videos set-ups. To evaluate the classifiers for each video set-up, we performed a ten-fold cross-validation by creating 10 different training/validation pairs by sliding the training data window by 10% each time. Then, for each of the training/validation pair, we performed 10 classificatory runs. We trained each run using the corresponding training set. Afterwards, we evaluated the classification using the rest of the dataset.

To determinate how the image resolution affects the performance of our algorithm, we tested different scales of resolution reduction between 1 to 16. The results shown on Table 2 were obtained using the video set-up I and Table 3 with the video set-up II.

Table 2: Comparison of accuracy, false alarm, missing alarm using different image resolutions on the video set-up I.

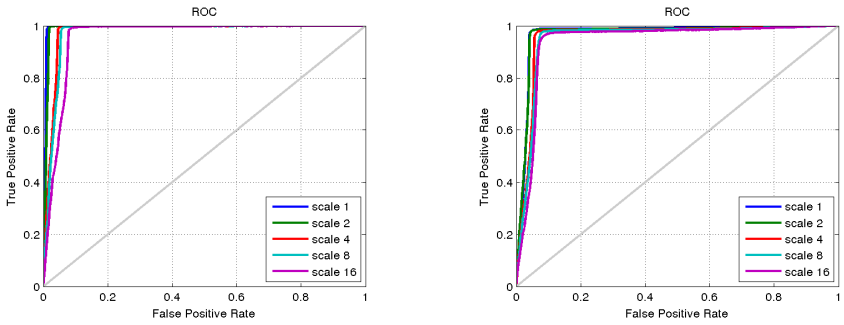
Image resolution	**Accuracy (%)**	**False Alarm (%)**	**Missing Alarm (%)**
640x480	99.59	0.16	1.12
320x240	99.52	0.17	1.37
160x120	99.17	0.41	2.06
80x60	99.11	0.67	2.41
40x30	98.17	0.77	4.93

The purpose of this experiment is to see the influence of the resolution in terms of accuracy, false alarm and missing alarm rate. For the first video set-up, we have a difference of 1.42% in terms of accuracy between the biggest and the smallest image resolution. With the second video set-up, this difference is around 2%. Fig. 9 shows the performance of our detector and the impact of the scaling.

An evaluation criterion is the processing time between the input image and the instant of detection. In these experiments, the average of processing time for a full image is around 7 s. This time is not acceptable on a real-time system. Table 4 shows the impact of scaling in the execution time. The main idea is to

Table 3: Comparison of accuracy, false alarm, missing alarm using different image resolutions on the video set-up II.

Image resolution	**Accuracy (%)**	**False Alarm (%)**	**Missing Alarm (%)**
752x480	98.28	1.02	2.78
376x240	98.06	1.12	3.19
188x120	97.86	1.32	3.40
94x60	97.41	1.47	4.29
47x30	96.32	2.08	6.126



(a) Video set-up I (b) Video set-up II

Fig. 9: Receiver operating characteristic (ROC) curves of two videos set-up with different images resolutions.

determine a good compromise between accuracy and execution time. Hence by reducing the resolution eight times for each dimension, the processing time is close to 0.3 s with fairly good accuracy.

Table 4: Comparison of time execution using different image resolutions on the video set-up II.

Image resolution	**Execution Time (ms)**
752x480	7097.10
376x240	1855.71
188x120	602.03
94x60	295.92
47x30	227.41

4 Conclusion

In this paper, we presented an embedded ground change detector aiming at warning walker's users before entering dangerous situations. This detector was

based on a multilayer perceptron processing the current frame and a number of preceding frames.

The obtained results demonstrated the possibility to implement a ground change detector in real-time on an embedded system. Moreover, image resolution reduction showed that image resolution reduction has small impact on accuracy loss, but with a high reduction factor of time.

Finally, the very promising results allow us to advocate the view that our detector could be possibly embedded in a device showing high accuracy in realistic situations.

Acknowledgments

This work has been developed as part of the EyeWalker project that is financially supported by the Swiss Hasler Foundation Smartworld Program, grant Nr. 11083.

We thank our end-users partner the FSASD, Fondation des Services dAide et de Soins Domicile, Geneva, Switzerland; EMS-Charmilles, Geneva, Switzerland; and Foundation Tulita, Bogota, Colombia.

References

1. United Nations. Population ageing and development. dec 2012. Accessed June 28, 2014.
2. Michel Bleijlevens, Joseph Diederiks, Marike Hendriks, Jolanda van Haastregt, Harry Crebolder, and Jacques van Eijk. Relationship between location and activity in injurious falls: an exploratory study. *BMC Geriatrics*, 10(1):40, 2010.
3. Andrew J Rentschler, Richard Simpson, Rory A Cooper, and Michael L Boninger. Clinical evaluation of guido robotic walker. *Journal of rehabilitation research and development*, 45(9):1281–93, 2008.
4. A. Frizera, R. Ceres, J.L. Pons, A. Abellanas, and R. Raya. The smart walkers as geriatric assistive device. the simbiosis purpose. *Gerontechnology*, 7(2), 2008.
5. Steven Dubowsky, Frank Genot, Sara Godding, Hisamitsu Kozono, Adam Skwersky, Haoyong Yu, and Long Shen Yu. Pamm - a robotic aid to the elderly for mobility assistance and monitoring: A "helping-hand" for the elderly. In *IEEE International Conference on Robotics and Automation*, pages 570–576, 2000.
6. Viviana Weiss, Severine Cloix, Guido Bologna, David Hasler, and Thierry Pun. A robust, real-time ground change detector for a smart walker. VISAPP 2014, 9th Int. Conf. on Computer Vision Theory and Applications,Lisbon, Portugal, 2014.
7. Pradeep Kumar and Vikas Gupta. Learning based obstacle detection for textural path. In *International Journal of Emerging Technology and Advanced Engineering*, volume 2, pages 436–439, 2012.
8. Otvio A.B. Penatti, Eduardo Valle, and Ricardo da S. Torres. Comparative study of global color and texture descriptors for web image retrieval. In *Journal of Visual Communication and Image Representation*, volume 23, pages 359 – 380, 2012.
9. OpenCV. Opencv. march 2015. Accessed March 20, 2015.
10. Gumstix. Caspa vl. march 2015. Accessed March 20, 2015.
11. Gumstix. Tobi. march 2015. Accessed March 20, 2015.
12. Gumstix. Overo waterstorm com. march 2015. Accessed March 20, 2015.