2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 21-23 September 2011, Guimarães, Portugal

Heading Change Detection for Indoor Navigation with a Smartphone Camera

Laura Ruotsalainen*, Heidi Kuusniemi* and Ruizhi Chen*
* Finnish Geodetic Institute/Department of Navigation and Positioning, Masala, Finland.
Email: laura.ruotsalainen@fgi.fi, heidi.kuusniemi@fgi.fi, ruizhi.chen@fgi.fi

***Abstract*— A comfortable and useful pedestrian navigation system is accurate, reasonably priced, easy to use and light to carry. Smartphones are attractive platforms for the navigation systems due to their small size, low cost and the fact that they are already carried routinely by many pedestrians. Pedestrian navigation is mostly needed in GNSS degraded and denied areas such as indoors and in urban canyons. Positioning in these environments is still a challenging task. Aiding from other location sensors is typically needed. Obtaining a reliable heading e.g. from a digital compass is one of the most challenging tasks indoors because of the environmental disturbances to the sensor measurements. Visual-aiding has been considered for solving this problem for the last few years. Most algorithms are however often too massive for a Smartphone with limited computing power and storage space. This paper presents a lightweight algorithm for solving the problem. It is based on vanishing points calculated from lines in consecutive images. Results from a field test have demonstrated that the performance of the algorithm in calculating the heading change is much better than that of the built-in digital compass in a Nokia Smartphone. The heading change can be detected with the proposed algorithm at about 1Hz frequency under the PC environment. It has potential to be adopted for a Smartphone platform.**

***Keywords*—Visual-aiding; Navigation; Smartphone; Indoor; Pedestrian**

I. Introduction

Pedestrian navigation is mostly needed indoors and in urban canyons where Global Navigation Satellite System (GNSS) observations are degraded or not available. To increase the availability and accuracy of position information in the GNSS-denied environments GNSS has to be aided with other systems and sensors. The use of inertial sensors for the aiding purpose has proven to be competent [1, 2]. The transition information of the pedestrian movement may be obtained with accelerometers and heading with, e.g., a digital compass. When these sensor measurements are integrated with a Pedestrian Dead Reckoning (PDR) algorithm and wireless network information the positioning of the user in a GNSS degraded environments is achieved [3-6]. The problem with the presented integration method is the errors in the inertial sensor measurements. The accelerometer results may be corrected to some extent with the PDR algorithm [7-8]. The errors in the digital compass are mostly due to electric objects and nearby ferrous materials in the navigation environment and are difficult to be corrected. Fig. 1 shows the heading errors obtained with a digital compass from a Smartphone (Nokia 6710). The heading was measured while walking along a corridor.

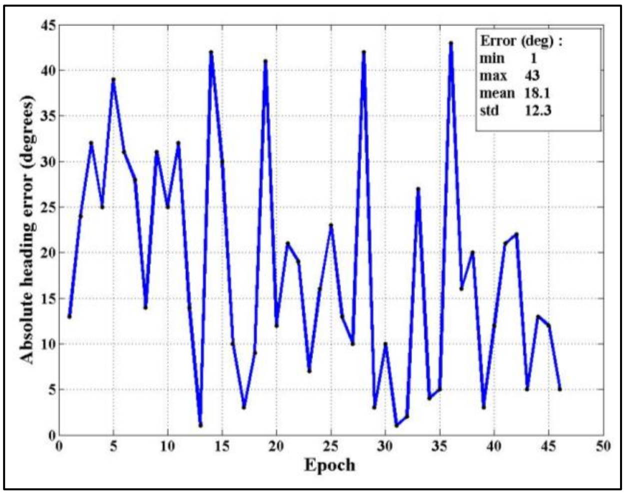

Figure 1: Absolute heading error obtained with a digital compass of a smart-phone.

Visual-aiding is a suitable method for correcting the pedestrian motion information. A camera is independent of the other sensors and doesn't suffer from disturbances induced by the navigation environment. It is also applicable in all environments as long as the lighting conditions are adequate even to some extent.

The long term objective of this research is to develop a seamless visual-aided pedestrian navigation system for a Smartphone. The system will integrate GNSS, wireless networks, inertial sensors and visual-aiding. All measurements will be integrated with a fusing filter to obtain an accurate user position. Consequently, the availability, continuity and accuracy of the navigation system will be enhanced.

A Smartphone is an attractive platform for pedestrian navigation due to its small size and low cost. Most Smartphones are also equipped with other systems needed for an integrated positioning solution including multiple sensors such as Global Positioning System (GPS) receiver, accelerometer, digital compass and camera as well as multiple Radio Frequency (RF) signal opportunities. However, real-time requirements of a navigation system set restrictions for using the existing visual-aiding algorithms in a Smartphone due to the limited computing power and storage space.

In this paper, the performance, accuracy and computing time, of basic visual-aiding algorithms for calculating the rotation angles of the camera along the x-axis is evaluated. The rotation computation is conducted by following the vanishing points calculated from images taken with a Smartphone.

II. CALCULATING THE HEADING CHANGE

The visual-aiding in the discussed system will be done by monitoring the motion of features matched in the consecutive images. Deriving information for matching from image is prone to errors caused mainly by the light conditions of the environment. Changes in camera orientation, viewing point and distance from features alter the shape perceived in images and make the matching difficult. Indoor environments where the navigation is mainly needed, like hallways of offices and public buildings, are poor with features. Also moving and far features give challenge for the methods. Fortunately, these surroundings have many parallel or perpendicular lines like edges of walls and doors [9]. For that reason selecting line features from images and calculating the vanishing points from intersection points of line pairs have been chosen to be the basis of the matching.

The purpose of this experiment is to obtain a comprehensive understanding of the algorithms needed and suitable for calculating the heading change of the camera in a Smartphone from consecutive images. For that, first, the lines in the images will be searched, their vanishing points calculated and from their motion the heading change angle of the camera derived.

A. Edge Detection

Fast changes of brightness in the image indicate edges of objects. This fact is also the basis of the Canny edge detector [10]. It’s an optimal algorithm for detecting the edges requiring low error rate of the calculations, the points to be well localized, meaning that the distance between the calculated location of the edge and the real one has to be minimal, and refusing multiple responses for an edge.

First the image is convolved with a Gaussian filter. This is done to reduce the noise in the image, which is the primary cause for disorder in edge detection. Gradient directions and magnitudes are calculated for each pixel. The edges are found with non-maximum suppression; a pixel having larger gradient magnitude than the neighbours along the gradient direction belongs to the edge. Since too many edges are obtained with the procedure, the weak ones have to be discarded with hysteresis. In hysteresis, two thresholds are chosen. All pixels above the higher one are decided to be included into an edge and below the lower one deemed not to. This leaves us with the optimal set of edges.

B. Separating the Lines from Other Edges

When all edges have been found using the Canny algorithm, the lines has to be distinguished from the other ones. A good method for that is the Hough transformation [11]. It detects which pixels belong to a line, which pixels belong to which line and how many lines may be found from an image in total [12].

The method is based on examining each edge pixel and giving a vote for all features possibly travelling through the pixel. The features gaining most votes are chosen to be the real ones.

Hough transformation is presented with parameters (ρ,θ) defining a line including the pixel (x,y) with (1)

$$\rho = x\cos(\theta) + y sin(\theta). \quad (1)$$

ρ is the radius of a line passing through the origin and normal to the line being detected and θ is the angle with the x-axis. The Hough space data is computed by calculating (1) for each pixel in the edge image with all possible values θ [13]. The data in the Hough space is then considered to be the votes for lines in the image.

C. Calculating the Central Vanishing Point

Vanishing points are good features to follow since they are invariant to position changes and projective transformations, meaning taking a 2-dimensional image of the 3-dimensional world [14]. Straight lines stay straight lines but parallel lines seem to intersect in one point in projective transformations. This point is the vanishing point.

Vanishing points may be calculated robustly using sophisticated algorithms like RANSAC [15, 9], Expectation-maximization algorithm [16], Hough transformation [13] or Levenberg-Marquardt [14] to name a few. The downside of these algorithms is their computational requirements that are evaluated to be too much for a Smartphone committing real-time calculations, as stated in the case of RANSAC in [17].

In the method presented in this paper, the vanishing points are calculated by voting for the intersection points of all line pairs [18] found with the Hough transform. The pixel getting most votes is the vanishing point. The procedure requires the intersection points to lie within the image space, but as the interest is only in finding the central vanishing points, the points where the lines converge in the horizon, this suits the existing needs well. Since even after the thorough pre-processing the images contain some noise the vanishing point has to be adjusted and the effect of outliers reduced. This is done with robust estimation using weighted means [19]. The corrected point is the mean of the initial pixel and its neighbours using the votes as their weights.

D. Calculating the Heading Change

Calculating motion from images requires calibration of the camera [20]. Calibration means calculating the camera matrix K (2) by resolving the intrinsic camera parameters, the focal length (f_x, f_y), coordinates of the principal point (u,v) and skew S.

$$K = \begin{bmatrix} f_x & S & u \\ 0 & f_y & v \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The position of the camera relative to the scene may be calculated using the vanishing points [21]. The scene is defined with the world coordinate frame, the XZ-plane being parallel to the floor and the Y-axis to the vertical edges of the doors. The Z-axis is parallel to the lines converging in the horizon. The three vanishing points v_x,v_y and v_z having the directions of the coordinate axes may be resolved with the camera matrix K and rotation matrix R using (3) [21]

$$V = [v_x \quad v_y \quad v_z] = KR. \tag{3}$$

While the three axes defining the camera coordinate frame are aligned with the axes defining the world coordinate frame, rotation is described with an identity matrix and the central vanishing point v_z lies in the principal point.

When the camera is rotated along the XZ-plane, that is changing the heading with an angle θ, change of position of the central vanishing point v_z is obtained using (4)

$$v_z = \begin{bmatrix} f_x sin\theta \\ 0 \\ \cos\theta \end{bmatrix}. \tag{4}$$

By resolving the change of positions of the central vanishing point in consecutive images, the rate of the heading change may be calculated.

III. Experiments

The suitability of the selected methods and algorithms in finding the vanishing points and the accuracy of calculating the heading change in one dimension was evaluated. This was done by attaching the Smartphone rigidly to a stationary platform and allowing the orientation change only along the x-axis. The environment was photographed by turning the camera 360 degrees and taking a photo every 5 degrees, resulting with 72 images in total. The reason for photographing the whole turn was to evaluate the sufficiency of the method for different scenes present in a common pedestrian navigation environment.

The camera was calibrated using the Matlab Calibration Toolbox [22]. The noise in the images was reduced using Gaussian linear filtering and the contrast of images enhanced. The Matlab's Canny edge detector that was used for finding the edges commits linear filtering also, but this was found insufficient.

Since the rotation of the camera was restricted to be only along the x-axis, merely the calculation of the central vanishing point lying inside the image plane was conducted.

A. *Test Equipment and Environment*

The digital still images used in the test were taken with Nokia's N8 Smartphone's camera having a resolution of 12 Mega pixels. The resolution of the images was reduced to 1280x960 to make them more manageable. To avoid the unwanted displacement of the camera, the images were captured using an autotimer implemented with Symbian software. This prevented the need to touch the phone during the experiment. The real heading change was constrained by using Edmund Optics' rotary stage.

All calculations were made with a PC using Matlab-software. Matlab's integrated Gaussian linear filtering, Canny edge detector and Hough transformation were used. The image contrast adjustment algorithm was made by Peter Kovesi [23]. The other algorithms utilized were implemented in this research process.

The tests were conducted in an office building in January 2011. The dark winter surroundings set also a lot of challenge for the image-aiding due to the lack of natural light. The camera was placed in a location enabling the photographing of very varying scenes. The environment consisted of two corridors with a field of view of 45 degrees in both.

Stairs with a rail having multiple lines lied between the corridors. One side of the scene consisted of a plane, wall and an elevator and a scene with a flower.

B. *Results*

All calculations needed for solving the heading change from the images including pre-processing the image, finding the edges and lines, calculating the vanishing points and the change of the view angle, took 1.1 seconds on the average, i.e. 79.4 seconds for 72 photos.

The significance of the pre-processing steps was tested by searching for the vanishing points by using either Gaussian filtering or contrast adjusting alone. The result was many incorrectly calculated points indicating the importance of both processes. The optimal parameters for the algorithms were also found by trial.

The mean error of the heading change detected in the first corridor was 1.3 degrees, while that for the second corridor was 1.8 degrees. Table 1 shows the statistics for the heading changes detected in both corridors. Fig. 2 shows the corresponding error plots.

While turning the camera into a scene with only a plane, wall and the elevator door, the calculations failed totally, producing change errors of 20-60 degrees. The calculations failed also while photographing a scene with a flower. The rail of the stairs having many lines with different slopes confused the computing and the heading change varied randomly. Fig. 3 shows an example of the vanishing points in corridor 1 (top) and corridor 2 (bottom) marked with a red circle. The blue lines are the lines found with Hough transformation.

TABLE 1
STATISTICS OF THE DETECTED HEADING CHANGES.

Statistics	min error (deg)	max error (deg)	mean error (deg)	std of error (deg)
Corridor 1	0.06	4.8	1.3	1.5
Corridor 2	0.04	6.4	1.8	2.3

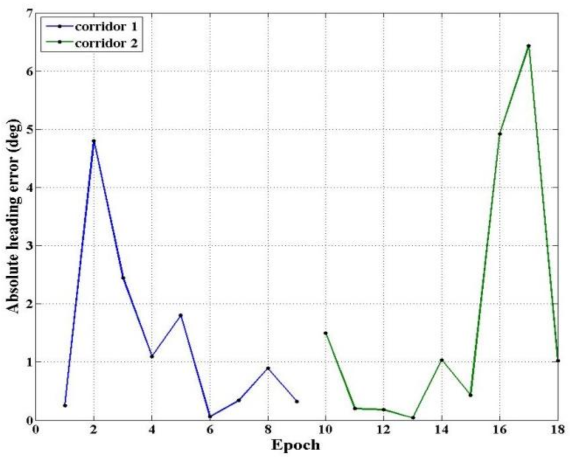

Figure 2: Visualization of the measured heading errors in degrees. Blue plot presents measurement errors obtained with the images from the first corridor and green plot same errors for the second.

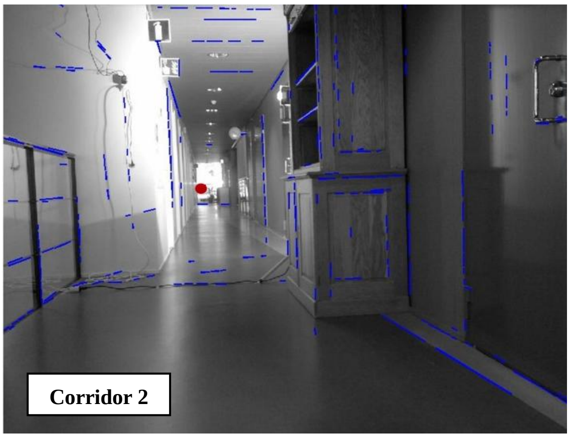

Figure 3: Image of corridor 1on top and corridor 2 on bottom. Blue lines were found with Hough transformation and the red spot shows the place of the vanishing point calculated with methods presented in this paper.

IV. CONCLUSIONS

An algorithm of detecting the heading change for indoors using a Smartphone camera has been developed in this paper. The lightweight algorithm in terms of computation power demands includes the following processing steps: image pre-processing, edge detection, line extraction, calculating central vanishing points and heading change detection. The performance of the algorithm has been evaluated via a field test carried out on the third floor an office building. The accuracy of the heading changes detected is about 1.5 degrees, which is much better than that of the built-in digital compass in a Smartphone (18.1 degrees for the same corridors). The computation time needed for each image is about 1.1 seconds in a PC environment. The results showed that the detection of the heading change is accurate to a satisfactory extent for pedestrian navigation while walking along a corridor, which is mostly the situation in real-life cases. However, the algorithm does not solve the problem for the cases when the images are taken facing a plane, like the elevator door, or a scene with plenty of features. These cases occur when the camera is headed off the corridors.

The experiments strengthened the decision of using the motion of vanishing points as basis for the visual-aiding in positioning. The algorithm developed can be adopted for indoor navigation.

REFERENCES

[1] J. Collin, Investigations of self-contained sensors for personal navigation, Doctoral Dissertation, Tampere University of Technology, 2006.

[2] G. Retscher, “Test and integration of location sensors for a multi-personal navigator”, the Journal of Navigation, vol. 60, pp. 107-117, 2007.

[3] R. Chen, Y, Chen, L. Pei, W. Chen, H. Kuusniemi, J. Liu, H. Leppäkoski, J. Takala, ”A DSP –based multi-sensor multi-network positioning platform”, Proceedings of ION GNSS, USA, pp. 615-621, 2009.

[4] S. Cho, C. Park, “MEMS based pedestrian navigation system”, Journal of Navigation, vol. 59, pp. 135-153, 2006.

[5] L. Fang et al., “Design of a wireless assisted pedestrian dead reckoning system – the NavMote experience”, IEEE Trans Instrum and Meas, vol 54, no. 6, pp. 2342-2358, 2005.

[6] S. Godha, G. Lachapelle, ME Cannon, “Integrated GPS/INS system for pedestrian navigation in a signal degraded environment”, Proceedings of ION GNSS, pp.2151-2164, 2006.

[7] Q. Ladetto, “On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering”, Proceedings of ION GPS, pp.1735-1740, 2000.

[8] R. Levi, T. Judd, Dead reckoning navigational system using accelerometer to measure foot impacts, US Patent US5583776, 1999.

[9] C. Kessler, C. Ascher, N. Frietsch, M. Weinmann and G. Trommer, **“**Vision-based attitude estimation for indoor navigation using vanishing points and lines”, IEEE ION/PLANS 2010, pp.310-318, 2010.

[10] J. F. Canny, “A computational approach to edge detection”, IEEE Transactions on Pattern Analysis and Machine Intelligene, vol PAMI-8, no 6, pp. 679-698, 1986.

[11] Hough P.V.C. Method and means for recognizing complex patterns. U.S. Patent 3069654. 1962.

[12] D.A. Forsyth and J. Ponce, Computer vision: a modern approach, Prentice Hall, 2003.

[13] J.M. Borkowski, M.J. Veth, “Passive indoor image-aided inertial attitude estimation using a predictive Hough transformation, IEEE / ION PLANS 2010, pp.295-302, 2010.

[14] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, second edition, Cambridge University Press, 2003.

[15] Fischler and Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”, Communications of the ACM, - 6, vol 24, pp. 381-395, 1981.

[16] G. Schindler and F. Dellaert, “Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments”, Proc. Conf. on Computer Vision and Pattern Recognition, pp. 203–209, 2004.

[17] M. Veth and J. Raquet, Two-dimensional stochastic projections for tight integration of optical and inertial sensors for navigation, report, Air Force Institute of Technology, Ohio, 2007.

[18] N.Lu, Z. Feng, “ Accumulative intersection space based corner detection algorithm”, International journal of pattern recognition and artificial intelligence, vol. 22, pp. 1559-1586, 2008.

[19] A. Jepson, D. Fleet, Foundations of Computer Vision, Robust estimation, lecture notes: http://www.cs.toronto.edu/~jepson/csc2503/index09.html , 2009.

[20] A. J. Davison, “Real-time simultaneous localisation and mapping with a single camera”, IEEE International Conference on Computer Vision, vol 2, pp. 1403 – 1410, 2003.

[21] A.C. Gallagher, “Using vanishing points to correct camera rotation in images”, IEEE CRV’05, pp.460-467, 2005.

[22] J-Y. Bouguet, http://www.vision.caltech.edu/bouguetj/calib_doc/

[23] P. D. Kovesi. MATLAB and Octave Functions for Computer Vision and Image Processing.

http://www.csse.uwa.edu.au/~pk/research/matlabfns/