

Mapping by seeing: wearable vision-based dead-reckoning, and closing the loop

Daniel Roggen, Reto Jenny, Patrick de la Hamette, Gerhard Tröster

Wearable Computing Laboratory, ETH Zürich, Switzerland
<http://www.wearable.ethz.ch/>
droggen@ife.ee.ethz.ch

Abstract. We introduce, characterize and test a vision-based dead-reckoning system for wearable computing that allows to track the user’s trajectory in an unknown and non-instrumented environment by integrating the optical flow. Only a single inexpensive camera worn on the body is required, which may be reused for other purposes such as HCI. Result show that distance estimates are accurate (6-12%) while rotation tends to be underestimated. The accumulation of errors is compensated by identifying previously visited locations and “closing the loop”; it results in greatly enhanced accuracy. Opportunistic use of wireless signatures is used to identify similar locations. No a-priori knowledge of the environment such as map is needed, therefore the system is well-suited for wearable computing. We identify the limitations of this approach and suggest future improvements.

1 Introduction

Wearable computers aims to empower users by providing them with information or support anytime and anywhere, proactively and in unobtrusive ways. By being “body-worn” wearable computers can sense the user’s state, such as his gestures, activities or his location. This *contextual awareness* is the key mechanism that enables wearable computers to support users proactively and in an unobtrusive way [1, 2]. The physical location of the user is an important contextual information that allows *location aware computing*, such as providing location-specific information in a wearable tourist guide [3], or learning a person’s daily activity patterns [4, 5].

Satellite positioning systems allow outdoor global localization, but they do not operate in difficult environments (e.g. indoor, urban canyons). Wi-Fi or GSM signals allow localization indoor as well as outdoor [6–9], but they generally need a map of the radio beacon location. Accurate localization is possible in environments instrumented with localization beacons (e.g. radio, ultrasound)¹. Yet these approaches are not well suited for wearable computing where the user can freely move in an open-ended environment: a-priori knowledge (map) or instrumentation should be avoided in this case.

Absolute localization (e.g. satellite) combined with *dead-reckoning* using body-worn sensors is a way to address this issue. Dead-reckoning in wearable computing has been investigated with combination of compass to estimate the heading direction and velocity estimation methods with e.g. step counting [11–13].

¹ For a review of localization methods, especially indoors, see [10].

In this paper we investigate for the first time in wearable computing a dead-reckoning method based on the integration of the *optical flow* from a camera placed on the front of the user to determine his *egomotion* (*vision-based path integration*). Vision-based dead-reckoning has the potential of being an inexpensive, low-power and highly integrated solution to dead-reckoning since a single camera is needed such as those used in cellular phones. This approach allows sensor reuse, e.g by using the same camera for vision-based HCI or even for taking pictures.

This approach is biologically inspired from insect flight and it has been used in mobile robotics for odometry, where it often benefits from controlled environments such as flat surfaces or good lighting conditions. In contrast, in a wearable setting the constraints on the environment are relaxed and noise in the optical flow induced by terrain geometry or camera angle variations caused by the movement of the user have to be considered. In this paper we translate known dead-reckoning methods using optical flow integration in a wearable computing application domain. Our objective is to investigate the challenges as well as the benefits and limitations of such an approach in this new application domain.

In order to cope with path integration errors, a path correction mechanism is implemented that “closes the loop” when identical locations are visited. Locations are identified from the signal strength of radio beacons opportunistically found in the environment. Yet the whole system does not need instrumented environments and operates without any a-priori knowledge, such as a map of the radio-beacons. It therefore fulfills the requirement for a scalable wearable mapping system.

This paper is organized as follows. In section 2 we explain how optical flow is computed and leads to egomotion. The setup is described in section 3. In section 4 we characterize the system in a variety of conditions and in section 5 we show path integration results for typical trajectories. In section 6 we discuss how path integration errors may be compensated by closing the loop when identical locations are visited. Finally we discuss the results in section 7 and conclude this paper in section 8.

2 Vision-based dead-reckoning

Vision-based dead-reckoning operates by integrating the optical flow (or image velocity) registered on a camera. The optical flow is a 2-D motion direction vector field which is the projection of the 3-D velocities of the environment surface points onto the image plane. The trajectory of a camera moving in the 3-D space (*egomotion*) can be reconstructed from the optical flow knowing the environment structure (e.g. geometry of obstacles or open environment). Two steps must be carried out: 1) find the optical flow in a sequence of images, 2) find the egomotion corresponding to this optical flow (i.e. dead-reckoning). Figure 1 illustrates the optical flow of typical motions.

Vision-based dead-reckoning has been studied in biology: insects such as blowflies or bees rely on optical flow to control flight and navigate [14]. The human brain also uses vision-based path integration to estimate travel distance and direction [15–17]. Optical flow has been used in mobile robots for navigation [18, 19], direction estimation [20] or odometry [21], and in cars for road-navigation [22].

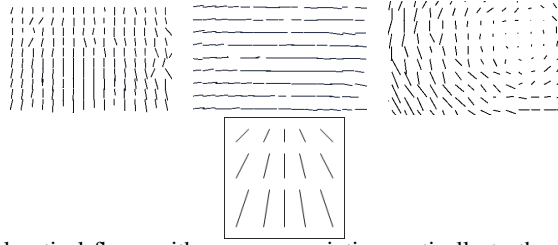


Fig. 1. Top: typical optical flows with a camera pointing vertically to the floor while moving forward (left), laterally (center), and rotating (right). The lines are the 2-D motion direction vector field on the camera. Bottom: schematic view of the optical flow with the camera moving forward and pointing with a 45° angle downwards.

2.1 Computing optic flow

A summary and comparison of the various techniques to estimate optical flow can be found in [23]. Here we use the Lucas-Kanade differential technique [24] because of its low complexity, with the modification proposed by [23]² to estimate accuracy of the optical flow estimation.

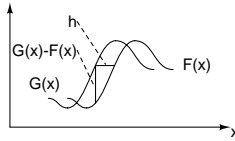


Fig. 2. The optic flow is the displacement h between $F(x)$ and $G(x)$ representing the light intensity in two successive frames.

Differential technique in one dimension In the one dimensional case the Lucas-Kanade method consists in finding the displacement h between two curves $F(x)$ and $G(x)$ representing the light intensity along the same line in two successive frames, where $G(x) = F(x + h)$ (fig. 2). The problem of finding h from $F(x)$ and $G(x)$ can be solved by a linear approximation of F in the neighborhood of x . For small h :

$$F'(x) \approx \frac{F(x+h) - F(x)}{h} = \frac{G(x) - F(x)}{h}$$

$$h \approx \frac{G(x) - F(x)}{F'(x)} \quad (1)$$

$$F'(x) = \frac{F(x+dx) - F(x-dx)}{2 \cdot dx}$$

This algorithm assumes h to be “small enough” in comparison to the frequency of $F(x)$ (see [24] for details). With fast displacements or low frame rate this condition may not be fulfilled.

² Sec. 2.1, paragraph referring to Lucas-Kanade’s method in that paper.

Differential technique in two dimensions In this work we implement exactly the Lucas-Kanade method as described in [23]. It is the 2D extension of the 1D approach described above. $I(x, t)$ is the intensity of a pixel at a position x at time t . We assume an image translation: $I(x, t) = I(x - v \cdot t, 0)$ with $v = (v_x, v_y)^T$ the displacement velocity. Equation (1) leads to the following *gradient constraint equation*³:

$$\nabla I(x, t) \cdot v + I_t(x, t) = 0 \quad (2)$$

$I_t(x, t)$ is the temporal derivative of $I(x, t)$ and $\nabla I(x, t) = (I_x(x, t), I_y(x, t))^T$ is the spatial derivative. A least-square fit under this constraint is performed in a small neighborhood Ω centered on the desired pixel. This yields a solution to v that consists in simple 2D matrix multiplications of the form $v = (A^T W^2 A)^{-1} A^T W^2 b$ with A , W and b being respectively the gradient of I , a weighting factor giving more importance to the constraints in the center of the neighborhood, and the temporal derivative of I . The mathematical steps are left in [23].

In this work Ω is a 5x5 pixel window, the derivatives are computed with three-point estimation, and the window function $W(x)$ is separable and isotropic with the effective 1-D weights (0.0625, 0.25, 0.375, 0.25, 0.0625). The eigenvalues $\lambda_1 > \lambda_2$ of $A^T W^2 A$ can be used to identify unreliable measurements. Here measurements with $\lambda_2 < 1$ are discarded. All other details are as in [23]⁴.

2.2 Computing egomotion

We consider 2 relevant degrees of freedom: the forward displacement (*translation*) and the angular rotation between successive frames⁵. They are accumulated to obtain the travelled path. Figure 3 top illustrates the system parameters. The camera with the horizontal and vertical aperture angle α_h and α_v is placed at a fixed height h and is looking downwards with an angle φ . A trapezoidal area of the floor with angle ψ is projected onto the camera image which has a resolution of res_x by res_y pixels.

During rotation, the optical flow is horizontal for pixels corresponding to the horizon; pixels mapping to the ground show a slight curvature. This curvature is negligible because the camera height and inclination map an area of the ground far away from the rotation center and we assume the optical flow vectors to have identical magnitude v_x along the horizontal direction. The rotation angle is computed as follows:

$$\Omega = \frac{res_x \cdot v_x}{\psi} \quad (3)$$

ψ is function of the camera angle φ : $\psi = 2 \cdot \arctan(\sin(\varphi) \tan(\frac{\alpha_h}{2}))$.

³ The link with the one-dimensional case is the following: $F(x) = I(x, t)$, $G(x) = I(x, t + \Delta t)$, $h = v_x \cdot \Delta t$, $F'(x) = I_x(x, t)$ the partial derivative along x .

⁴ Sec. 2.1, paragraph referring to Lucas-Kanade's method in that paper.

⁵ In our scenario the camera is fixed to the body and therefore more than 2 degrees of freedom (DoF) apply. While in principle the six DoF of the camera motion can be reconstructed from the optical flow [25], we consider here the 2 major DoF as a way to assess the performance of the system with the simplest motion model.

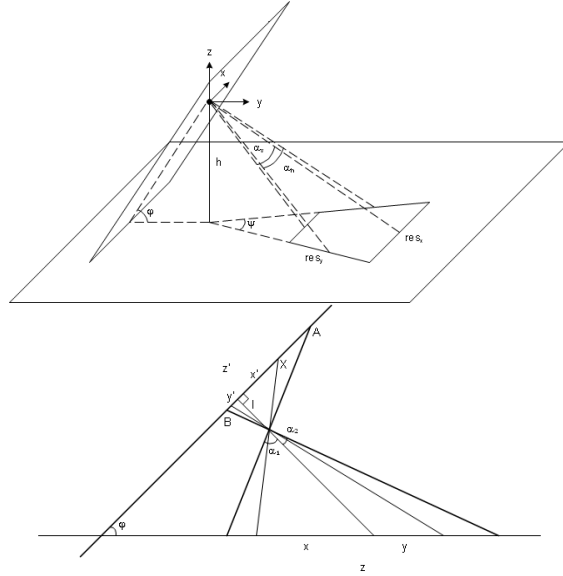


Fig. 3. Estimation of egomotion from optical flow. Bottom: tilted camera looking at the floor and mapping of ground points x and y onto the image x' and y' .

During translation with the camera moving forward and pointing downwards the x -component of the velocity vectors can be neglected without loss of information. The y -component of the velocity vectors have different magnitude at the top and bottom of the frame (see fig. 1 bottom) and must be taken into consideration. The mapping between pixels and the ground is found according to figure 3 bottom. The mapping between the optical flow and the ground motion is given in equation (4). α is the half aperture angle of the camera (in the figure α_1 and α_2 are the aperture angle measured from the point where the normal axis trough the camera crosses the image plane; in practice $\alpha_1 = \alpha_2 = \alpha$). z is the displacement on the ground in meters, h the camera height, z^* is the optic flow measured in pixel at position p_s (pixels) in the image, and p_m is the vertical image size in pixels. p_f is a factor to convert between pixel and meters: $p_f = (2\tan(\alpha))/p_m$. The overall translation is the average of z computed for each motion vector with equation (4).

$$z = \frac{h}{\cos(\varphi)} \left[\frac{\tan(\alpha) - p_f p_s}{\cos(\varphi) + \sin(\tan(\alpha) - p_f p_s)} + \frac{\tan(\alpha) + p_f p_s - p_f p_m + p_f z^*}{\cos(\varphi) - \sin(\tan(\alpha) + p_f p_s - p_f p_m + p_f z^*)} \right] \quad (4)$$

In order to improve accuracy the optic flow is computed and averaged on 10 frames; only optic flow vectors which satisfy the reliability conditions are considered ($\lambda_2 > 1$). The rotation and displacement is computed with equations (3) and (4). From the starting position $(0, 0)$ with heading 0° , the path is integrated by combining the rotation and displacement vectors frame by frame.

3 Setup

Images are acquired from a fixed focal length Logitech QuickCam Pro 5000 webcam which has a maximum framerate and resolution of 30fps and 640x480 pixels. The results presented below are obtained at 30fps and 120x160 pixels. A Java program allows offline and batch video sequence processing with many adjustable parameters. A C filter plugin for Virtual Dub allows real-time processing while capturing the video and it superposes the integrated path on the video frames. In both cases uncompressed videos are acquired and processing is done on the luminance channel.

A “synthetic” setup is used for characterization purposes in good lighting conditions (high contrast). The camera is moved smoothly over a textured surface at a fixed height and with a fixed angle α . A ground covered with newspaper pages is used, as well as other floor structures (fig. 4) Ground truth values for speed and rotations were obtained from markings made on the surface at known distances and angles. The speed and rotation speed is obtained by counting the number of frames between the appearances of the markings. In the real-world setup the camera is fixed on the chest of the user, pointing forward and downwards, so that only the ground is visible in the frame (height 1m45, approximate angle $\alpha = 45^\circ$).

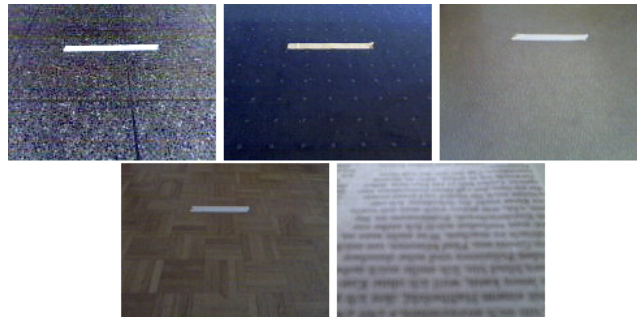


Fig. 4. Floors: concrete, carpet, linoleum, parquet, and newspaper on the floor.

4 Characterization

The system is characterized in the synthetic environment by moving the camera straight for a fixed distance of 30cm. Figure 5 left illustrates the distance travelled by the camera as estimated from the optical flow for various heights and effective speeds ($\alpha = 45^\circ$). Figure 5 right illustrates the same with various floor types ($h = 54cm, \alpha = 45^\circ$). The distance tends to be underestimated at higher speeds and a higher camera tends to allow faster speeds. The underestimation is caused at higher speeds by the image that tends to blur and the “small displacement” condition that is not respected. With $h = 90cm$ the distance is constantly underestimated due to the blurred images acquired from the camera which has a focal length optimized for short distances (i.e. the fine textures disappear). The ground type has only limited impact on the results and similar results were achieved with the dark carpet and the lighter parquet textures.

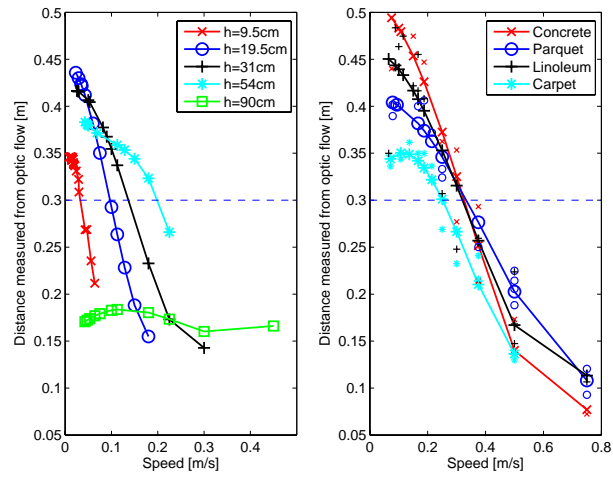


Fig. 5. Camera travel distance estimated from the optical flow for various effective speeds and camera heights (left) and floor types (right).

Rotation estimation is characterized in the same way by rotating the camera smoothly at various rotation speeds until a 90° rotation is achieved. The camera angle is $\alpha = 45^\circ$. The rotation angle of the camera as estimated from the optical flow is illustrated in fig. 6 left for various camera height and in fig. 6 right for various floor types ($h = 9.5\text{cm}$ in this case). The angle tends to be underestimated with increasing angular velocities. Furthermore even for small angular velocities the underestimation is already high. The reasons are similar as before: the small displacement assumption is not respected and the image tends to blur with higher rotation speeds. Even low rotation speeds correspond to a large image translation in terms of pixels. As a consequence the small displacement assumption is not respected even for low rotation speed, which explains the underestimation.

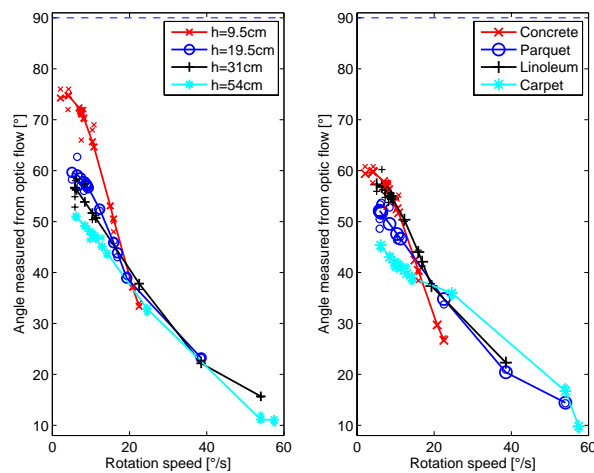


Fig. 6. Camera rotation angle estimated from the optical flow reported effective rotation speeds and various camera heights (left) and floor types (right).

The results may be improved by a mathematical compensation based on the system characterization. The incremental angle and displacement estimated from the optic flow is corrected at each time step by a factor linearly proportional to the angular speed or the velocity. The correction function is determined from analysis of the system's behavior at various rotation speed and camera angles. If some parameters are fixed (e.g. fixed camera angle) this correction function can be further optimized for this case. Figures 7 and 8 illustrate the system characteristics without (left) and with (right) mathematical compensation. The real rotation and translation is 90° and 0.3m , the camera height 53cm .

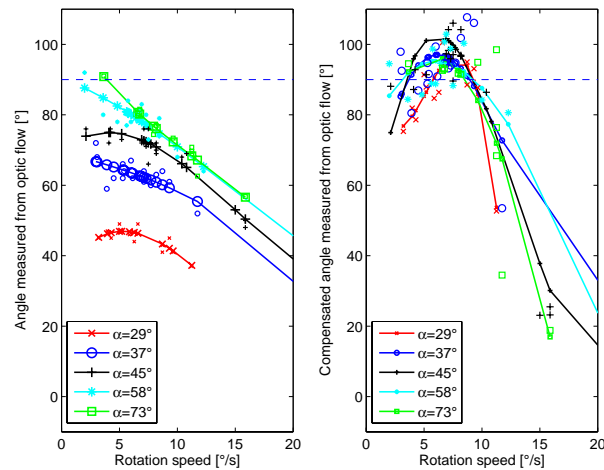


Fig. 7. Original (left) and mathematically compensated (right) angular measurement at various speeds and camera angles.

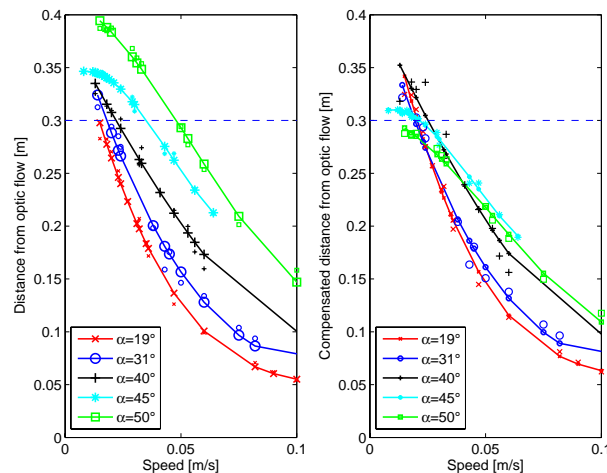


Fig. 8. Original (left) and mathematically compensated (right) distance measurement at various speeds and camera angles.

5 Path integration results

Figure 9 illustrate the path integration result in the synthetic environment (floor textured with newspaper) when moving along a square and a circle at a slow ($<0.02\text{m/s}$, $8\text{-}10^\circ/\text{s}$) and higher speed. Camera parameters are $h=9.5\text{cm}$, $\alpha = 45^\circ$. Dashed lines indicate the reference trajectory. The distance is well approximated from the optic flow, but inaccuracies in the estimation of the angle, especially with the higher speed, evidence the “closing of the loop” problem: the trajectory estimated from the optical flow ends away from the starting position, although the start end end point match in the reference trajectory. Figure 10 illustrates a more complex trajectory: a 1/80th scale of our laboratory building⁶.

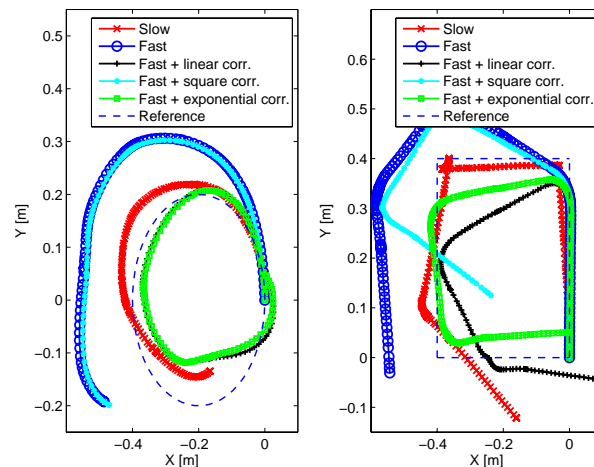


Fig. 9. Path integration while moving along a circle (left) and a square (right). Slow and fast refers to the motion speed. Corrected paths use the closing of the loop.

Tests were performed by wearing the camera on the chest (height 145cm, angle of approximately 48°). Figure 11 illustrates the real and estimated speed profile with a user walking straight on 5.5m on a concrete floor (fig. 4 left). The estimated travelled distance is 5.17 meters (relative error of 6%). The instantaneous speed measured by the optical flow reveals the periodicity of the footsteps. Speed under- or over-estimation depends mostly on the camera angle, which may be caused by the user bending. Figure 12 illustrates the trajectory of a user walking along a square of 10m sides (concrete floor). The estimated travelled distance was 43.4 meters whereas the real distance was 38.8 meters (slightly below 40m because of the rounding of the corners), corresponding to a relative error of 12%. This error is higher than in the straight walk, and may be caused by the inaccuracies introduced by the rotation. The rotations are however strongly underestimated. This is explained by the high speed at which humans rotate

⁶ Tests in the real building were not successful: no reliable motion vectors could be extracted due to the highly uniform texture of the carpet. A scaled synthetic environment was used instead with floor textured with newspaper.

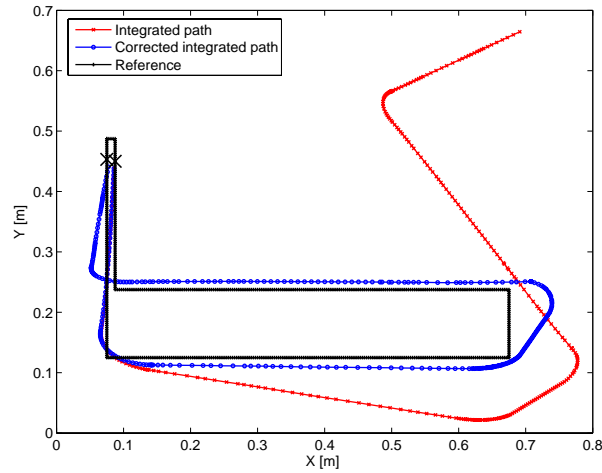


Fig. 10. Path integration while following the corridors of a laboratory building.

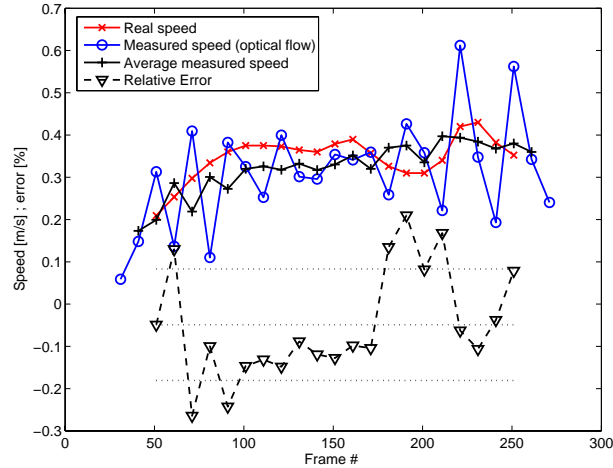


Fig. 11. Estimated speed of a user with a chest-worn camera walking straight for 5.5m. Dashed lines indicate the average and standard deviation of the relative speed error.

while walking ($30^\circ/s$ to $90^\circ/s$) which far higher than what the current system can handle as characterized before.

6 Closing the loop

When a user's start and end point of a trajectory coincide, the errors accumulated by dead-reckoning lead to the "loop closing" problem: the end point mapped from dead-reckoning is likely not to correspond with the start point. This problem may be addressed by recognizing locations previously visited and correcting the previously mapped trajectory in order to "close the loop" by matching the two visited locations. In other words, we use the knowledge that two visited locations are identical to correct a-posteriori the drift resulting from the integration of the optical flow.

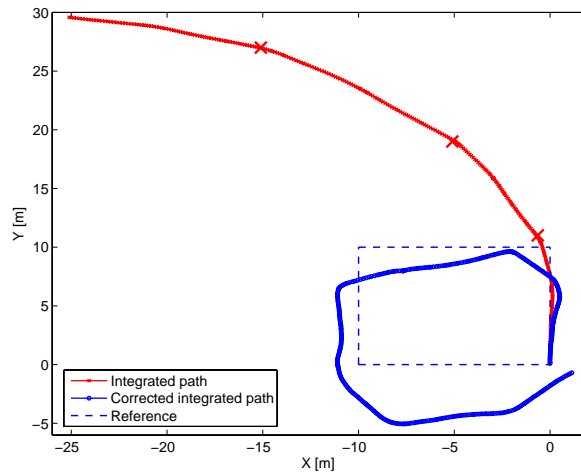


Fig. 12. Path integration of a user with a chest-worn camera walking along a square with 10m sides. The crosses on the integrated path indicate when the user rotates in the corners of the square.

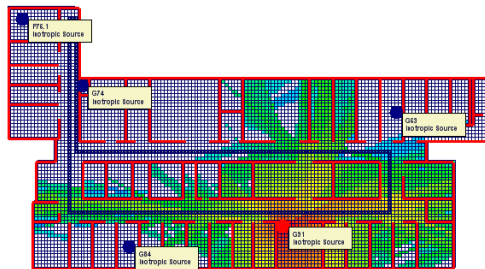


Fig. 13. Signal propagation.

Path integration errors may stem from errors in the estimation of the motion vector magnitude or orientation. Here we consider only the orientation that is the main source of errors⁷. We close the loop by iteratively increasing the angle difference $AngleDiff$ between successive motion vectors until the path is closed using a “corrected” angle difference of the form $AngleCorr = f(AngleDiff)$. Here faster rotations are underestimated more than slow ones. We tested linear, power and exponential correction functions and found the best function to be exponential: $A \cdot (e^{AngleDiff \cdot B} - 1)$ with $0.05 \leq A \leq 0.5$ and $0.5 \leq B \leq 1.5$ found by testing. A search algorithm finds the correction parameters that minimize the distance between two identified points of the trajectory through exhaustive search in the parameter space.

⁷ We assume that the magnitude and orientation of motion vectors is under- or over-estimated in a systematic way. Previous results show that the errors in the estimation of the angular rotation dominate over the errors in the estimation of the translation distance. For this reason we correct here only the orientation of the motion vectors with a formula that attempts to compensate for the under- or over-estimation of the angular rotation. This approach benefits any type of trajectory and not only circular or rectangular path.

Figure 9 illustrates the correction functions when the camera moves along a circle and a square (same conditions as in section 5). Figures 12 and 10 use the exponential correction and illustrate the improved dead-reckoning with the closing of the loop algorithm.

In order to recognize previously visited locations, characteristic landmarks must be identified and compared against previously memorized ones (e.g. visual landmarks [26]). We show preliminary results in simulation where the “wireless signature” (signal strength of nearby WiFi stations) is used for this purpose⁸. We simulate the signal propagation of the 5 WiFi stations of the building where the trajectory of fig. 10 would take place using the Radio Propagation Simulator of Radioplan⁹. We record the signal intensities on the whole floor with a grid spacing of 0.25m and on the 617 points of the trajectory where at least two WiFi station were in range. The signature matching achieves a $< 5m$ average localization accuracy in this simulation (nearest match). This accuracy indicates how close the start and end point of a loop must be for them to be matched. In the trajectory of fig. 10, we compared the 20 first meters of the trajectory with the 20 last meters of the trajectory, and found that the two points marked by the cross are the closest (0.65m) and should trigger the closing of the loop as illustrated.

7 Discussion

Alternative dead-reckoning methods mostly rely on step count integration, using accelerometers and compass [11–13]. In the experiments carried out here the system measured the travelled distance with an accuracy between 6% to 12% depending on the path. This compares well to the accuracies reported in [12], which vary between 1.3% and 29% depending on the body location of the system. Step count integration requires to know the length of the step (a-priori definition or inferred from the acceleration signal). Here we do not require this knowledge since the optical flow is considered; as such gait dynamics has in principle no influence on the performance of the system, which would work even if the user were on a bike or scooter.

A benefit of this vision-based approach is that a single camera sensor is needed, which may also be used for other wearable computing activities such as taking pictures or vision-based HCI. In particular in [27] a chest-mounted “FingerMouse” camera system was investigated which captures hand motion as an input for wearable computers.

⁸ Visual landmarks would make the best use of the available camera. Recognizing visual landmarks is challenging and computationally intensive. Instead we use simpler wireless signatures. Our objective is to assess the implications of “closing the loop”. The exact way in which previously visited locations is determined does not affect the algorithm correcting the trajectory nor the results. This approach to the use of wireless signatures is similar to [8, 9]. In these work the wireless signature is used for accurate localization, by relying on a previously learned “wireless map” of the environment that indicates typical wireless signatures at each position. Here we avoid this wireless map measured a-priori. We simply reference current wireless signatures to past ones, without mapping them to absolute positions. Therefore no a-priori knowledge of the environment is needed.

⁹ <http://www.radioplan.com>. We used the free version which simulates a limited number of reflections and penetrations.

Further work may combine HCI and localization in a single chest-mounted camera system.

In this method we evidenced several trade-offs between parameters (camera placement, image parameters, motion) and system performance. In theory the system is not sensitive to floor type or color as long as contrast is perceived. In practice the spatial frequency of the floor is linked to the resolving power of the camera. Performance degrades at higher rotation/translation speeds. This is caused by the assumption of small displacement of the differential technique that is not verified at higher speeds. This assumption is also linked to the spatial frequency of the textures [24]. This can be compensated by increasing the framerate (results not presented here confirm this) or placing the camera higher. A larger ground area is projected in the image frame with a higher camera, which allows for faster motion (reduction in the spatial frequency of the textures and thus the small displacement condition is valid for higher speed). A too high height leads to difficulties in obtaining the motion vectors (e.g. blurred images due to the fixed focal length, or uniform texture such as carpet which mostly contains high spatial frequency textures). This may be compensated by a higher frame resolution.

Alternatively another algorithm to obtain the optical flow may be used, such as looking for features in successive frames. Block-matching as used in MPEG video encoding may be used for this purpose. By adjusting the block search neighborhood, the small displacement constraint may be alleviated and the system may operate at faster speeds. The computational requirements of block-matching are higher than those of the differential method, but optimized instruction sets (e.g. MMX) allow to compute these operations in real-time on any 1+GHz computer.

Other sources of errors stem from the hypotheses of our approach: the camera parameters (angle, height) are known, and the camera motion is only a combination of forward motion and rotation. Errors are introduced when these are not verified. Changes in camera height and angle lead to an over- or under-estimation of the speed, with the camera angle being the most important parameter. Uneven floor or tilted floor as well as body motion (leaning forward/backward) are similar to a change in camera angle. Torso rotation (e.g. looking sideways while walking) or bending sideways affect the system since they introduce optical flows that are not expected. The footsteps are easily distinguished from the instantaneous speed estimate (fig. 11) but are not a source of error since they are averaged over time. Interestingly this may lead to a visual approach to count the footsteps. Further improvement may be obtained by optimizing the mathematical compensation method for a particular camera angle or surface structure, at the loss generality. Alternatively image processing may detect typical surfaces to optimize the compensation parameters.

The system may be improved with additional sensors. A sensor giving the camera angle (e.g. accelerometer) may allow to take into consideration the instantaneous camera angle while integrating the path. A compass may allow to improve the rotation estimation accuracy from the optical flow. The floor geometry is important, yet recent findings suggest this approach may work even in extreme terrains [28]. Finally, a more complex model may allow to reconstruct the full 6D egomotion from optical flow [29, 18, 25]. In principle optical flow may thus be used in a wearable setting without con-

straint on the user motion, however it remains to be investigated what would be the tradeoffs of such algorithms.

Dead-reckoning runs in real-time at 160x120, 30fps on a 1.6GHz Intel Centrino CPU without any implementation optimization: all the computation is done in floating point without use of vectorial instructions. Fixed-point optimization and use of vectorial instructions is likely to allow a speedup of more than one order of magnitude and therefore allow higher resolution. A hardware optical flow implementation may also be investigated. Low-power analog VLSI chips directly give the optical flow at a high frame rate [30, 31]. Digital hardware implementations of block matching may also allow low-power and high-speed processing (e.g. in [27] the authors present a block-matching approach to stereovision for HCI; the same block matching structure may be used to compute the optical flow).

We presented a method of opportunistic use of wireless signature to compensate for the errors accumulated by dead-reckoning in simulation. Future work will have to test this in reality. There are however a number of results suggesting that this approach may be applicable [8, 9].

In summary, the system introduced here requires sufficient floor texture and contrast to operate. This precludes the use of such approach on uniform surfaces (depending on camera resolution), or in low lighting conditions (e.g. night). With the current egomotion model, the system performs best when the user is walking straight and performing rotations without significant sideways bending. Closing of the loop needs sufficiently distinct wireless signatures in the environment and is expected to work best in environments with dense mesh of wifi stations. With a low density of wifi stations the wireless signatures become less accurate (e.g. with a single isotropic wifi station the same signature is sensed at identical distance from the station regardless of the bearing).

8 Conclusion

The objective of this paper was to investigate how vision-based dead-reckoning methods may be applied in wearable computing. We were appealed by the fact that a single miniature and low-power camera may be used for this purpose. We introduced, characterized and tested a method for dead-reckoning based on the integration of the optical flow obtained from an inexpensive chest-mounted camera. Results show that the distances estimated from the optical flow are accurate for a walking subject (6-12% relative error) and comparable to methods integrating steps. The rotation tends to be underestimated due to the high angular speed at which users turn in relation to the framerate (30 fps). This calls for higher framerates in order to improve rotation estimation accuracy. We proposed a method to improve the path integration accuracy by opportunistically using wireless signatures to detect already visited location. The integrated path is corrected at this moment by “closing the loop” so that the visited locations match. We showed that the accuracy of the trajectory mapping was greatly enhanced in this way. A localization method for wearable computing needs to be “mapless” so that it can be used in any environment. This vision-based dead-reckoning method combined with the opportunistic use of wireless signatures does not require any a-priori map or knowledge about the environment. It thus satisfies this requirement and can be used anywhere.

The application in wearable computing is complex due to the free motion of the user and the unconstrained nature of the environment in which he may walk in. Our re-

sults are preliminary: further tests in realistic scenarios are needed, and we do not claim to a final vision-based dead-reckoning solution. Still, our results allow us to evidence a number of challenges, limitations and trade-offs which may shape future work. We highlighted the links between system accuracy, camera and environment parameters, and the user motions. Higher frame rates and higher camera resolutions improve the system accuracy, but improved characterization, improved optical flow models, as well as additional sensors may also improve the performance. We believe that there is more to this approach than what was presented here, as is also suggested by the numerous literature describing the use of optical flow in mobile robotics. Our results evidence topics for further research. **Technical improvements** such as the use of higher resolution camera, higher framerate, and autofocus, may improve the system accuracy. A **motion model** with additional degrees of freedom may allow to better cope with motions typical of wearable setting. Comparative analysis may evidence more accurate or computationally-efficient methods to derive **optical flow**. Finally, we believe **sensor fusion** may best address the shortcomings of the current approach by using additional modalities. For instance the use of magnetometers to improve estimation of rotations may be investigated, as well as the use of accelerometers or inclinometers to take into account the camera angle in the optical flow model.

References

1. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context awareness. Technical Report GITGVU-99-22, Georgia Tech (1999)
2. Lukowicz, P., Junker, H., Staeger, M., von Bueren, T., Troester, G.: WearNET: A distributed multi-sensor system for context aware wearables. In Borriello, G., Holmquist, L., eds.: *UbiComp 2002: Proceedings of the 4th International Conference on Ubiquitous Computing*, Heidelberg, Springer (September 2002) 361–370
3. Simcock, T., P., H., Thomas, B.H.: Developing a location based tourist guide application. In Johnson, C., Montague, P., Stekettee, C., eds.: *Conferences in Research and Practice in Information Technology*, Australian Computer Society (2003) 177–183
4. Patterson, D.J., Liao, L., Fox, D., Kautz, H.: Inferring high-level behavior from low-level sensors. In Goos, G., Hartmanis, J., van Leeuwen, J., eds.: *Proc. of the 5th Int. Conf. on Ubiquitous Computing*, Heidelberg, Springer (2003) 73–89
5. Ashbrook, D., Starner, T.: Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing* 7(5) (2004) 275–286
6. Haeberlen, A., Flannery, E., Ladd, A.M., Rudys, A., Wallach, D.S., Kavraki, L.E.: Practical robust localization over large-scale 802.11 wireless networks. In: *Proc. of the 10th annual international conference on Mobile computing and networking*, New York, ACM Press (2004) 70–84
7. Cavalieri, S.: A novel approach for localisation based on wi-fi. In: *Proc. of the 3rd Int. Conf. on Industrial Informatics (INDIN)*. (2005) 234–239
8. Otsason, V., Varshavsky, A., La Marca, A., de Lara, E.: Accurate gsm indoor localization. In Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H., eds.: *Proc. of the 7th Int. Conf. on Ubiquitous Computing (UbiComp 2005)*, Heidelberg, Springer-Verlag (2005) 141–158
9. La Marca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G., Schilit, B.: Place lab: Device positioning using radio beacons in the wild. In: *Proc. of Pervasive Computing*, Heidelberg, Springer (2005) 116–133

10. Hightower, J., Borriello, G.: Location systems for ubiquitous computing. *IEEE Computer* (2001) 57–66
11. Lee, S.W., Mase, K.: Activity and location recognition using wearable sensors. *Pervasive Computing* **1**(3) (2002) 24–32
12. Randell, C., Djiallis, C., Muller, H.: Personal position measurement using dead reckoning. In: Proc. of the 7th IEEE Int. Symposium on Wearable Computers (ISWC'03, IEEE Computer Society (2003) 166–173
13. Schindler, G., Metzger, C., Starner, T.: A wearable interface for topological mapping and localization in indoor environments. In Hazas, M., Krumm, J., T., S., eds.: Proc. of the 2nd Int. Workshop on Location- and Context- Awareness, Heidelberg, Springer (2006) 64–73
14. Schilstra, C., Van Hateren, J.H.: Blowfly flight and optic flow I: thorax kinematics and flight dynamics. *The Journal of Experimental Biology* **202**(11) (1999) 1481–1490
15. Ellmore, T.M.: Human path integration by optic flow. Master's thesis, Department of Psychology, The University of Arizona (2004)
16. Frenz, H., Bremmer, F., Lappe, M.: Discrimination of travel distance from 'situated' optic flow. *Vision Research* **43**(20) (2003) 2173–2183
17. Frenz, H., Lappe, M.: Absolute travel distance from optic flow. *Vision Research* **45**(13) (2005) 1679–1692
18. Srinivasan, M.V., Chahl, J.S., Zhang, S.W.: Robot navigation by visual dead-reckoning: inspiration from insects. *International Journal of Pattern Recognition and Artificial Intelligence* **11**(1) (1997) 35–47
19. De Souza, G.N., Kak, A.C.: Vision for mobile robot navigation: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **24**(2) (2002) 237–267
20. Kröse, B., Dev, A., Groen, F.: Heading direction of a mobile robot from the optical flow. *Image and Vision Computing* **18**(5) (2000) 415–424
21. Nagatani, K., Tachibana, S., Sofue, M., Tanaka, Y.: Improvement of odometry for omnidirectional vehicle using optical flow information. In: IEEE/RSJ Int. Conference on Intelligent Robots and Systems. (2000) 468–473
22. Giachetti, A., Campani, M., Torre, V.: The use of optical flow for road navigation. *IEEE Transactions on Robotics and Automation* **14**(1) (1998) 34–48
23. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. *International Journal of Computer Vision* **12**(1) (1994) 43–77
24. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. of the 7th Int. Joint Conf. on Artificial Intelligence (IJCAI). (1981) 674–679
25. Nagle, M.G., Srinivasan, M.V., Wilson, D.L.: Image interpolation technique for measurement of egomotion in 6 degrees of freedom. *Journal of the Optical Society of America A: Optics, Image Science, and Vision* **14**(12) (1997) 3233–3241
26. Newman, P., Ho, K.: Slam- loop closing with visually salient features. In: IEEE Int. Conf. on Robotics and Automation (ICRA). (2005) 635–642
27. de la Hamette, P., Tröster, G.: Architecture and applications of the fingermouse: a smart stereo camera for wearable computing HCI. *Personal and Ubiquitous Computing* (online first) (2007)
28. Campbell, J., Sukthankar, R., Nourbakhsh, I.: Techniques for evaluating optical flow for visual odometry in extreme terrain. In: Proc. of Intelligent Robots and Systems (IROS). (2004) 3704–3711
29. Srinivasan, M.V.: An image-interpolation technique for the computation of optic flow and egomotion. *Biological Cybernetics* **71**(5) (1994) 401–415
30. Liu, S.C.: A neuromorphic avlsi model of global motion processing in the fly. *IEEE Trans. on Circuits and Systems* **47**(12) (2000) 1458–1467
31. Stocker, A., Douglas, R.: Analog integrated 2D optical flow sensor with programmable pixels. In: Int. Symposium on Circuits and Systems ISCAS. (2004) 121–138