Computer Vision and Image Understanding 149 (2016) 126–145

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

An assistive haptic interface for appearance-based indoor navigation

Jose Rivera-Rubio[a,*], Kai Arulkumaran[a], Hemang Rishi[b], Ioannis Alexiou[a], Anil A. Bharath[a]

[a] Department of Bioengineering, Imperial College London, London, United Kingdom
[b] Department of Electrical and Electronic Engineering, Imperial College London, London United Kingdom

ARTICLE INFO

Article history:
Received 18 April 2015
Revised 21 February 2016
Accepted 25 February 2016

Keywords:
Human navigation
Assistive technology
Localisation
Mobility
Indoor navigation

ABSTRACT

Computer vision remains an under-exploited technology for assistive devices. Here, we propose a navigation technique using low-resolution images from wearable or hand-held cameras to identify landmarks that are indicative of a user's position along crowdsourced paths. We test the components of a system that is able to provide blindfolded users with information about location via tactile feedback. We assess the accuracy of vision-based localisation by making comparisons with estimates of location derived from both a recent SLAM-based algorithm and from indoor surveying equipment. We evaluate the precision and reliability by which location information can be conveyed to human subjects by analysing their ability to infer position from electrostatic feedback in the form of textural (haptic) cues on a tablet device. Finally, we describe a relatively lightweight systems architecture that enables images to be captured and location results to be served back to the haptic device based on journey information from multiple users and devices.

1. Introduction

For most people, navigating familiar environments might seem a very natural task that usually involves travelling along a path that we have previously visited and learned. At other times, navigation might require us to follow a new and unseen path, and require planning and evaluation of possible directions of movement. Traditionally, finding our way in unfamiliar environments required certain skills such as map or compass reading. External cues also helped: signs, landmarks, directions from other people, etc. Recently, due to the emergence of smartphones and other 'wearables', we have devices that gather data, interpret it and provide tools to assess one's position; planning a route is almost immediate. Because these tools are increasingly available on a single device: the problem of navigation in outdoor contexts is reduced to the simple act of following the indications of a navigation App on a mobile device, or a "SatNav" [66].

The nature of the navigation problem in unfamiliar indoor environments might seem simpler than outdoors. Despite the fact that, in global terms, we are restricted to moving over a small region of the surface of the earth, some buildings—universities, museums, government buildings, shopping centres, airports and so on—can have vast internal dimensions. Even for sighted users, the frustration of getting lost in indoor environments can be accentuated, perhaps because the immediacy and efficacy that is achieved outdoors with automotive and naval navigation systems cannot be easily matched with the existing tools for indoor navigation.

Regardless of whether one is outdoors or indoors, navigation in sighted humans relies heavily on the sense of vision [34,68]. When vision is deteriorated or deprived, a person's ability to navigate—particularly in unfamiliar settings—is greatly diminished. Indeed, a significant proportion of individuals who experience sight loss late in life find navigation in unfamiliar environments challenging. This might be a key contributor to the fact that more than half (55%) of the blind and partially sighted in the UK rarely venture outside of their homes [77].

Despite demonstrations of promising technology on small scales of usage [40,42,45], the white cane remains the most widely used navigational aid. Helping blind and partially sighted people to navigate in unfamiliar environments is a particular challenge. There are several reasons for this, including immaturity of localisation technology in indoor settings, the cost of installing customised localisation technology, and the challenges associated with keeping mapping information up-to-date.

As for the case of vehicular navigation, it is likely that the solution to indoor navigation lies within not one, but rather a collection of approaches that work together. For reasons of both precision (a statistical argument, based on acquiring independent measurements) and redundancy (an engineering principle), several possible sources of localisation data, methods of user interaction and algorithms should be developed and evaluated separately. In

* Corresponding author.
E-mail address: jose.rivera@imperial.ac.uk (J. Rivera-Rubio).

http://dx.doi.org/10.1016/j.cviu.2016.02.014
1077-3142/

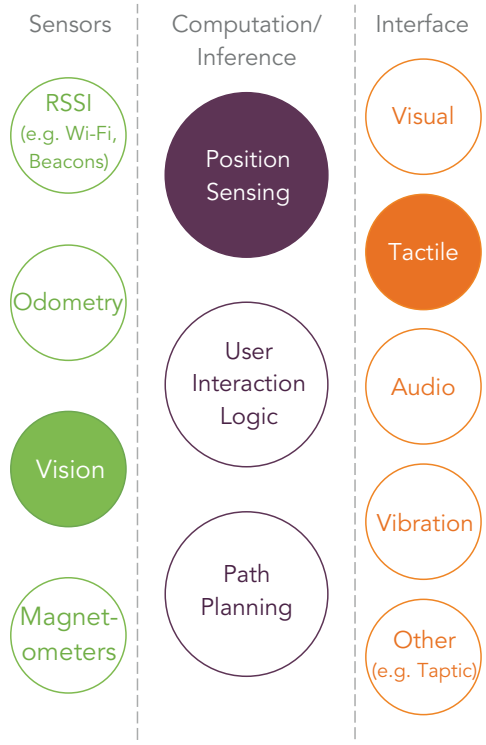


Fig. 1. The solid circles indicate the remit of this paper. We do not suggest that either visual sensing, tactile feedback or knowing one's position on a map solves the indoor navigation problem. In this paper, we have deliberately selected one sensing technique, one mechanism of feedback and one inference technique of the many redundant components and subsystems that one would wish to have in a robust navigation device. Evaluating components in a combinatorial manner allows redundant and robust modules to be created systematically, with component-level performance characterisation.

this paper, our intention is to take one combination of sensor, one type of inference method and one type of user interface to provide navigation information (see Fig. 1).

Selecting elements of a complete solution in this way allows us to isolate and characterise the performance of individual components. To enable location cues to be acquired from many users of the same space, we focus on the use of low-resolution video cameras as navigation sensors. This is combined with relatively simple algorithms, and the principle of using data from many journeys as an implicit reference system for one's location within a building. A full navigation system must compute possible paths as well as sense location, but here we focus on inference of position. Finally, people tend to show a range of preferences when using interface devices. We selected a haptic device to convey positional information, and evaluated the accuracy of this device for indoor localisation. Taking the three components together, our prototype system provides a user with the ability to determine where they are on a map or floor plan, a sufficiently useful task in its own right.

2. Background on assistive devices: accessible technology

2.1. The impact of sight loss in navigation

Clearly, the ultimate goal would be to prevent people losing their sight or to heal sight loss. In the absence of these achievements, both governmental agencies and charities have identified that support for independent living for people with visual impairment is a priority. In the UK, for example, the leading sight loss charity, the Royal National Institute of Blind people (RNIB), has identified two key aims [58] (slightly paraphrased for clarity):

- more people should be able to make journeys safely and independently;
- more people should achieve independence through the use of information technology and mobile technologies.

The importance of navigation for visually impaired people features prominently because of its impact on a person's independence. Studies have found that less than half (45%) of people with visual impairment go out every day, a fifth do not go out more than once a week, and nearly half (43%) would like to go out more often [22,58]. Additionally, a 2012 survey carried out during an accessibility event organised between the RNIB and Android London revealed that the most desired mobile application among members of the blind and partially sighted community would be a navigation application with access to important information such as signage or information panels, found mainly in written formats [59].

An engineering solution that supports navigational autonomy of the user is needed. In the next sections, we will describe some studies that have approached the navigation problem from different perspectives.

2.2. Non vision-based solutions for assistive navigation

2.2.1. Classical aids

The two principal navigation aids for visually impaired people remain the guide dog and the white cane. In addition to being good navigational aids in complex environments, guide dogs have been recognised as being a source of companionship, helping to combat isolation. However, the cost of training dogs can be high and the potential to get lost remains, particularly along unfamiliar routes. Also, obstacles that are above the height of the dog (such as low-hanging branches) can present a hazard [43].

The other highly successful piece of navigation technology for visually impaired people is the white cane [60]. Whilst it is known to allow more independence, it does not provide information on navigation on a spatial scale much greater than a stride length [42]. There are some navigation scenarios, such as those involving environments that are unfamiliar or too complex, which are avoided by some white cane users. Examples of these include walking a route for the first time, using public transport, approaching a building entrance or door to public transport. In particular, public transport usage remains extremely low among visually-impaired people, with just 11% of blind or partially sighted travellers boarding a train or a bus regularly [52].

2.2.2. Radio frequency systems

The latest systems for accurate outdoor navigation rely on information provided by satellite-based navigation systems, most commonly the Global Positioning System (GPS). Systems based on GPS have changed the current concept of outdoor navigation. Some significant attempts have been developed to target the needs of visually impaired people. For example, the Sendero Group's *Mobile Geo* [63] system uses GPS to provide position and navigation directions through an accessible keyboard and a speech synthesis interface. *BlindSquare* for Apple mobile and tablet devices takes GPS a step further by using crowdsourced data for points of interest (via integration with *Foursquare* services and data) and *OpenStreetMap* for fine-grained street information [45]. However, even such customised systems lack the information sources or signal availability indoors to be used by people with visual impairment. For example, the signal strength that reaches devices indoors is relatively weak—of the order of a tenth of a femtowatt [75]—and often unstable. Because of this, a system that is reliant on GPS indoors would provide subpar navigation, give a poor user experience, and potentially even compromise safety.

Several other projects have attempted to provide navigational information indoors based on radio frequency technologies. According to the RNIB [77], RFID, Wi-Fi and Bluetooth radio technologies can provide both accuracy and coverage indoors. Additionally, body sensors employing ZigBee Radio Signal Strength Indicators (RSSI) [21] have demonstrated the feasibility of wearable sensor networks to provide navigation information. Such networks usually require some form of infrastructure to be deployed throughout buildings. Signal transmitter locations then need to be tested, associated with indoor mapping information, and subsequently maintained; a process that can be costly.

Finally, *Drishti*, an integrated indoor/outdoor navigation system, has been proposed [54]; this uses Differential GPS (DGPS) for outdoor positioning and an ultrasound positioning device for indoor location measurements. Although reporting sub-meter localisation errors, the indoor subsystem requires the deployment of ultrasound transmitter "pilots", and the user has to carry ultrasound beacons and specialised hardware, making *Drishti* a technically feasible but costly and currently impractical prototype.

2.3. Tactile interfaces for the blind and partially sighted

Today, there are two common sensory modalities that we use to understand our surroundings: vision and, crucially for visually impaired users, hearing. Much load is already placed on these sensory channels, and they may reach a point of saturation in busy environments. Therefore, it is prudent to use another sensory modality to convey information, and one that has been investigated since the 1800s is touch.

The very earliest documented example of converting visual cues into tactile ones appears to be the Elektroftalm, created using a single block of selenium [17]. The photoconductivity of this block of material was used to convey a sensory stimulus to the foreheads of blind people, allowing them to distinguish between dark and light. The simplicity of this approach is in stark contrast to later attempts to transfer optical cues into tactile. For example, Bliss [14] combined a tactile stimulator with an optical sensor to allow the blind to understand their surroundings. The image found by the optical sensor fell onto a 12×12 phototransistor array and used a one-to-one mapping onto tactile stimulators. Each illumination of the phototransistor led to a vibration on the corresponding tactile stimulator. Only crude images were produced and it was found that not many visual objects were recognised reliably. However, Bliss identified that the results of this experiment may have been subject to defects in the intensity of responses in the piezoelectric units.

In more recent times, there have been many advances in tactile technologies. Users can experience tactile feedback in displays through several cues, including piezoelectric sensors [51], shape memory alloys [73], micromachined devices [39] and air jets [10]. In addition, there are promising directions around the use of electrorheological fluids (fluids that respond to electric fields by changing their viscosity) [51]. A common class of technique under exploration is vibrotactile displays. These use a combination of microlinear electromagnetic actuators and piezoelectric ceramics. However, they do not seem to convey the frictional forces which visually impaired users are attuned towards when exploring objects with their fingers [28]. More recently, [29] explored a piece of technology designed to enable visually-impaired users to find the distance to an object, such as a wall, by distance sensors worn on the head. In this arrangement, tactile cues were provided through vibrating patterns in a hand-held device [29].

Tactile technologies are also being considered for several applications for a wider range of users (including those with and without visual impairment), ranging from providing cues for pilots in flight [66] to computer mice [4]. In addition, there have been attempts at implementing tactile technologies into consumer devices. For example, Motorola found that out of 42 subjects, 35 preferred having a combination of vibrotactile feedback and visual cues [15], an outcome supported by previous research [53]. Poupyrev and colleagues claimed that tactile interfaces were a "peripheral awareness interface": they provide sensory stimulation on a subconscious level, thereby taking cognitive load off the user. Further uses of tactile technologies include training for surgery, in which it is sometimes necessary for a surgeon to be able to function under circumstances in which there is limited visibility [31].

For our purposes, the Senseg^{TM} tablet is an apt modern example of a tactile display that allows a user to feel something akin to frictional forces. The Senseg^{TM} device passes a low current to an isolated electrode that applies a small attractive force to the skin of the finger. By modulating this force, a device can convey the sensation of different textures. This is a rather significant advancement on the mechanical piezo solutions used by [14]. For the experiment described in this work, a Senseg^{TM} tablet will be used to test the information delivered as the result of an appearance-based location query (see Section 6.3).

2.4. Computer vision for navigation

2.4.1. Methods for inferring geometry: SfM and SLAM

The extended use, minimal cost and increasing quality of modern cameras have brought the use of visual information for assistive devices a step closer to reality. The inclusion of high-quality cameras in mobile devices, such as phones and tablets, has boosted the familiarity of users with both the use of cameras and the idea of camera-based mobile applications.

One outcome of the proliferation of these cameras is the increased interest and use of Structure from Motion (SfM) algorithms. These are able to infer 3D models of city regions [2] from photographs taken by visitors to popular city landmarks. With such images acquired from the Internet [65], bundle adjustment can be used to reconstruct the 3D information about buildings within well-photographed locations, in addition to the camera pose of every photograph. Hile and colleagues also crowdsourced location information [30] through geotagged images from Flickr, then used Snavely's SfM algorithm [65] to perform camera pose estimation. Using models of a scene constructed using bundle adjustment, the position and pose of a camera from a sequence of new photographs taken with a mobile device can be used as a source of "visual" navigation information [72]. However, bundle adjustment is an iterative error minimisation algorithm, and its computational load is still large for real-time use at scale. Furthermore, it is not entirely clear how the geometric information that is acquired from such models could be updated as aspects of a scene change.

Recently, RGB-D devices, producing both colour images and depth information, have shown promise for robotics. For example, [5] make use of an inexpensive RGB-D sensor to detect obstacle-free paths; both depth and colour information are used to infer the presence or absence of obstacles. Potentially, this is a vital feature for visually impaired people, as it extends the range of obstacle detection provided by the traditional white cane.

Another important branch of vision-based navigation techniques is to be found in robotics. Visual Simultaneous Localisation and Mapping (SLAM) [24,37,46] provides a real-time reconstruction of the scene by using either stereo cameras (stereoSLAM) or a single camera (monoSLAM). Although SLAM is often described for its ability to infer a geometric model of a scene, it also estimates a camera trajectory. The combination of the two is a powerful source of navigation information. In addition, during subsequent journeys along the same route, geometric information can be refined and used in camera pose estimates.

Some visual SLAM algorithms provide a navigation method suitable for autonomous robots [37]. Such techniques normally rely on static features from the scene that are subsequently matched before the trajectory is estimated. However, in real life conditions, many detected features are dynamic, since they correspond to objects or elements of the scene that are moving (e.g. people). Additionally, SLAM algorithms, particularly visual monoSLAM [19], rely on the optic flow induced by ego-motion in order to infer scene geometry and build up a map. In the presence of significant additional motion within the scene, SLAM algorithms can begin to fail.

Amongst other relevant techniques that are SLAM-based is the work of Alcantarilla et al. [6,8]. They incorporate dense optical flow estimation into visual SLAM in order to improve the performance of algorithms in crowded and dynamic environments by detecting the presence of objects that are moving relative to the world coordinate system. Additionally, they developed a fast vision-based method to speed up the association between visual features and points in large 3D databases [7]. This approach consists of learning the visibility of the features in order to narrow down the number of matching point correspondence candidates. Also of note is the modification to SLAM to address the common problem of scale-drift [67].

A SLAM-based solution that is closer to our approach was suggested by [9], who used a standard EKF-SLAM approach to track SIFT features. The SIFT features are simultaneously used to provide semantic information (i.e. object recognition for obstacle avoidance and path recognition) about the environment. Apart from the fact that the authors do not test vision in isolation (that is, they enforce tracking), their method's caveat is that instead of building a visual path with crowdsourced image descriptions or sensor signatures, they impose a known constraint on the "true pathway" for the location of the detected features, and also as a prior for the tracking. This constraint limits the scalability of the method; perhaps more importantly, it does not explore the accuracy of a location estimation system based on crowdsourced visual information.

One of the more recent developments is LSD-SLAM [23], a semi-dense tracking and mapping method that performs well in indoor settings. Instead of keypoints and descriptors, LSD-SLAM uses semi-dense depth maps for tracking by direct image alignment. This is a remarkable step forward, as the semi-dense maps allow computationally lighter frame-to-frame comparisons, to the point where odometry can be performed on a modern smartphone [61]. This system, as with most SLAM methods, relies on accurate camera calibration and initialisation routines, and best results are often achieved under specific camera and lens combinations, such as monochrome global shutter cameras with fish eye lenses [35].

2.4.2. Appearance-based methods for inclusive visual navigation

Appearance-based methods attempt to provide localisation without keeping track of the coordinates of the robot/user or landmarks in metric space.

For example, [26] presents a Bag-of-Visual-Words (BoVW) approach to provide a "qualitative" recognition of the room their robot is in. In summary, a set of features is extracted, a dictionary is built, and using a voting system as a classifier, a label (class) is assigned to each room. This is arguably not a SLAM method, as there is no simultaneous localisation and mapping that is comparable to mainstream SLAM methodologies. In fact, there appears to be no mapping beyond a table of the rooms visited. Secondly, the method is only partially a topological localisation technique, since there is no modelling of the relationships between rooms. However, one important contribution is the incremental approach to building a database. Filliat, rather than relying on a given set of categories (the rooms), creates new categories on the fly based on a decision made from the statistics of the visual words contained within an image.

FAB-MAP [18] and its popular open-source implementation [27] have been considered the state-of-the-art appearance-based method for robot navigation. FAB-MAP relies on a dictionary of Bag-of-Visual-Words constructed from a database of Speeded Up Robust Features (SURF) features extracted from location images to provide matching between previously visited places as well as a measure of the probability of being in a new, unseen location. The arrival of SeqSLAM [44] brought additional robustness to the FAB-MAP paradigm. By enforcing sequential constraints to the image matching front-end they were able to improve on OpenFABMAP, especially in challenging situations such as nighttime or during rain. To date, however, such appearance-based methods in a SLAM context are more commonly used at large spatial scales in order to address the loop-closure problem; its application is less common in indoor spaces when operating at smaller spatial scales. In addition, there is little evaluation of the effect of matching ambiguity when using appearance-based techniques.

The closest approach to the one described in the present work is perhaps that of [62]. Schroth and colleagues made use of a purely appearance-based method to provide localisation from image sequences acquired with mobile devices. However, their experiments are closer to those of the object categorisation community, where tests are performed in a batch fashion and performance is reported as precision-recall or receiver operating characteristic (ROC) curves. Our view, however, is that localisation error distributions are also a good illustrative metric of performance. In addition, their use of a 360° camera to produce a database for training somewhat constrains the effectiveness of the system to the quality and richness of this database, almost ensuring poor results when certain conditions are not captured in the database, as demonstrated by [44]. Crowdsourced training instances provide information that can largely solve this issue [57]. Schroth et al. [62] also suggested and developed a prototype client–server application, although using a different approach to the one we explore in this paper: instead of performing all computation on the server, they pre-select some relevant visual words that are sent to the client for matching. Whilst the work of Schroth et al. is relevant to that reported here, the lack of an assistive context makes the challenges different.

However, there are some examples of assistive applications of appearance-based methods. Ali and Nordin [9] developed an appearance-based method that uses SIFT features in order to construct a weighted topological map of the environment stored in a modified electronic white cane. During a query, the cane submits SIFT features that are matched with the ones in the database. The degree of similarity or "weight" between the matched images allows direction instructions to be conveyed based on previous knowledge about the environment. Nguyen et al. [47] combined SLAM with FAB-MAP to develop a mapping and visual localisation system based on constructing a route map that contains a database of images together with an odometry model. Their appearance-based method is limited to what FAB-MAP offers (SURF features), but to improve indoor reliability, they introduced markers that are easier to distinguish along the route. [48] introduced a standard Kalman filter SLAM approach to track the detected features in order to improve the robustness of location estimation for a robotic aid for visually impaired users.

Previous work [57] compared the performance of different appearance-based techniques for indoor localisation, extending the use of these methods beyond loop closure, allowing for positioning on its own. Average absolute position errors of as low as 2 m were reported using an approach based on matching images against crowdsourced journeys made along indoor corridors. This technique has some precedence in the literature, with [40] using a database of keyframes registered with 2D positions and orientations that were later used in an "online" mode for servicing queries

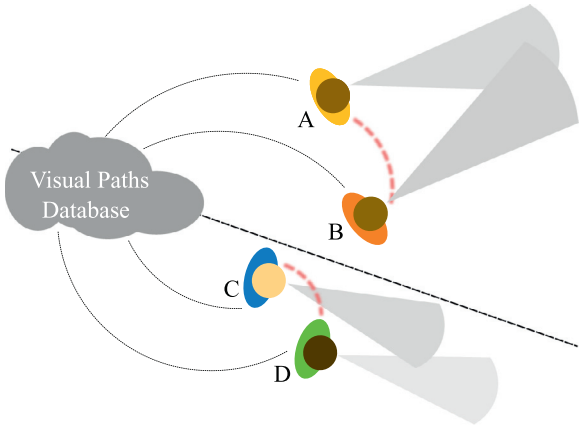


Fig. 2. Crowdsourcing indoor journeys ("visual paths") from multiple users. Users *A* and *B* make the same journey at different points in time, but can associate their journeys through storing their visual paths on a server; other users *C* and *D*, make different journeys, but again can associate their experiences with each other.

that consisted of GIST and SURF descriptors. A state estimator based on a Hidden Markov Model was also used for state prediction, and to enforce spatio-temporal consistency. The authors, however, did not appear to test their system "in the wild": for example, the database images were post-processed to reduce for motion blur. For this, they used an external inertial motion unit to capture information about the roll and pitch angles of motion.

In the next sections, we describe a BoVW approach to estimate a user's position during indoor navigation by using images acquired from either hand-held or wearable cameras. Position is estimated with respect to the distance travelled along one-dimensional paths consisting of ambiguous corridors; this presents a difficult use case for techniques such as SLAM, as will be shown in later sections.

2.5. Getting data into a navigation system: crowdsourcing

As we saw in Section 2, crowdsourced data is already enriching location information through social networking and personalised place recommendations (e.g. Foursquare); and through collaborative maps (e.g. *OpenStreetMap*). Crowdsourcing sensor data from mobile phones is providing a myriad of applications, from detecting traffic congestions [12] to mapping the real mobile network coverage based on thousands of individual signal strength readings [50]. More relevant to the present work is the study by [74] where indoor localisation is provided by matching inertial and magnetometer readings to sensor signatures stored in a database of crowdsourced data from previous users traversing the same space. We are adding vision to this hypothesis, and we propose two different scenarios for crowdsourcing of visual data:

a) visual data, together with ground truth positioning, is incorporated into mapping information as part of an accessibility measure. There are tools available for this process that standardise and accelerate the acquisition of visual data (see, for example, [32]).
b) individual users contribute recordings of their indoor journeys from wearable cameras and provide some contextual information and ground truth via a Web or mobile application.

These two scenarios are compatible in the sense that users should be able to enrich public indoor maps through crowdsourcing tools and benefit from the availability of this data through accessible Apps installed on their mobile/wearable devices. An illustration of this scenario is depicted in Fig. 2.

Therefore, we consider first the role of an *appearance-based* technique for using low-resolution images from a hand-held or wearable camera as both a source of query information and a source of database (mapping, localisation) information. Images are compared in order to establish position, and this can be seen as a means of externalising visual memory [69].

3. System overview

A key contribution of this work is to explore the feasibility and usefulness of an App that provides a haptic interface for appearance-based indoor localisation. In Fig. 3, we illustrate the concept: a blind or partially sighted user wants to travel from a point *A* to a point *B* in a building. They launch an App which starts collecting images from the camera of the Senseg^{TM} tablet or from a wearable camera paired with the tablet. These images are sent to the server, which estimates the location of the user based on an appearance-based visual localisation algorithm. The estimated location is sent back to the user's device where it is interpreted and conveyed in the form of a haptic cue over a pre-loaded floor plan of that part of the building. The device can also show visual feedback for sighted users, as illustrated in Fig. 7.

3.1. The data sources

As suggested in Section 1, both the floor plan and a database that contains previously acquired views (see Fig. 2) must be available. The former should be (politically) easy to acquire, particularly if it is considered as a means toward supporting accessibility. One option for the latter is described in Section 6.1, but note that, as shown in Fig. 1, the necessary data can be acquired or inferred through a variety of techniques.

A key concern for such databases, and particularly those that would seek to acquire image or video information upon which to base navigation services, is the sheer quantity of information that would need to be stored, acquired and transmitted. This is where the choice of processing technique can make a significant difference. Ideally, we should aim to capture, store and process visual information using as small an image size as is practically feasible to permit location recognition. We base the following calculations on uncompressed video, as most algorithms that generate visual descriptors for object recognition currently operate outside of the compressed domain. Furthermore, the range of descriptor implementations that we are able to choose from is dramatically increased by working in the image domain. A 15 s video clip acquired at normal walking speeds equates to just over 20 m of distance in real space. UMTS 3G mobile can run at up to 48 kB/s, enough to perform uncompressed image transfer for a 208 $\times$ 117 pixels greyscale image—a location query—within 1 s. However, 1 MB/s—the speed of EV-DO [13]—or 9.4 MB/s—the uplink speed of LTE 4G—is more than sufficient for acquiring crowdsourced low-resolution video data through streaming. Beyond bandwidth, the amount of storage required for these videos should also be considered. We found that 60 s of low-resolution video footage at 25 fps occupied between 2 and 3 MB when compressed, with 20 such journeys, each of 60 s duration, coming in at under 50 MB. These minimal storage costs are only achievable if the spatial resolution of the image data is sufficiently low—significantly lower than that used by current computer vision techniques for localisation. We observed that, by eye, we were able to determine the location of a person in a building based on 208 $\times$ 117 pixels greyscale video recordings from an equipped wearable camera. The question was whether localisation could also be achieved by an algorithm at such low resolutions.

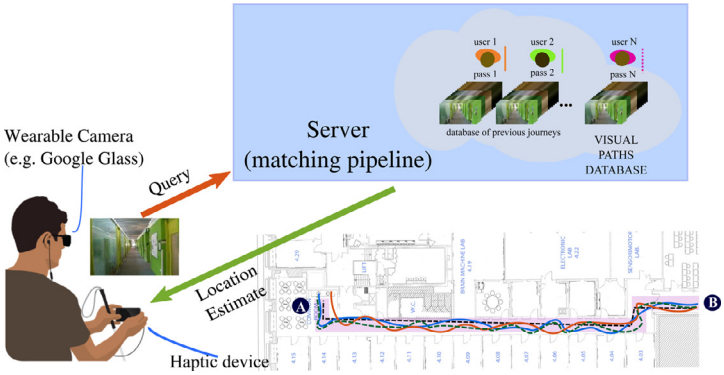


Fig. 3. Illustration of the usage scenario. The App installed on the user's tablet submits queries taken from a coupled wearable camera or the tablet's camera. A server sends location feedback, conveyed via tactile cues over a floor plan scaled to fit onto the device screen. The user is depicted with earphones to illustrate the use of multiple feedback interfaces; audio feedback was not implemented in this work, but see [15] for an example.

3.2. Algorithm choice

In previous work, image patch descriptors were evaluated for their ability to discriminate location [56]; in this paper, we use two methods of image description: one designed to extend gradient-based indexing techniques, permitting both scalable location indexing and the representation of textures via a haptic device (Gabor-based), and one widely available technique with many implementations (SIFT).

We describe the algorithm choice for visual processing in some detail in Section 4. For the RSM dataset [55], the inference of geometry and camera odometry can operate in real-time on a mobile device. However, running LSD-SLAM at the resolution required for tracking to succeed can quickly deplete a device's battery. Furthermore, the use of several uncalibrated devices, a likely scenario when trying to crowdsource information, poses a challenge to LSD-SLAM. Perhaps more importantly, utilising several sources of navigation information adds robustness, and allows routes to be updated at low cost. This requires a repository of journey sequences; the repository is therefore a key part of the architecture—and of the algorithm—used in creating the prototype App.

An indexing process considers all frames from multiple journeys, using this to build a custom dictionary that can be used to quickly search for matching frames, and which can apply distance measures between candidate BoVW descriptors. Metrics can be used to pass information back to a server about relative distance based on previous journeys along the same route. We expand on this in Section 4.

The data repository consists of the frames at both the original and compressed resolutions, and binary files for processed data (descriptors, dictionaries and encoded visual words). The sizes of the binary files were small enough that we could efficiently store and retrieve the data from both file systems on a single disc and an open source distributed file system (GlusterFS) spread over three Ubuntu servers.

3.3. Interface device

We chose to use the tactile interface of a prototype version of an electrostatic device, the Senseg™ tablet. This tablet is a customised version of a Google Nexus 7 Android tablet, and can provide a fairly rich tactile experience. In order to provide a scalable and real-time localisation service to a person, we utilised a standard client–server model, with a customised App on the Senseg™ tablet as the client. A HTTP server was implemented with Node.js, which acted as a proxy for calling the localisation code. This generic, modular design allows us to both extend the HTTP server's functionality, for example to include capturing of data for the dataset via another phone App, or to change the implementation of the HTTP server or localisation code independently.

We note that not only does the HTTP-based approach allow for relatively fast communication over a building's Wi-Fi network, but other network communication protocols are often blocked in institutional networks. The server can also be extended to use HTTPS and can operate on either non-standard or standard ports (80/8080 for HTTP and 443 for HTTPS).

4. Visual processing for localisation

We used an appearance-based search pipeline, shown in Fig. 5. It consists of a feature extraction process, followed by dictionary creation and encoding. However, it is designed to support experimentation with all stages of processing: in previous work we followed the same pipeline to evaluate several feature extraction techniques [56], with a mix of single-frame and spatio-temporal descriptors. We focussed on dense methods, as these were found to perform better for indoor navigation [56] in the datasets which we used for testing. Additionally, the distance metrics used in this paper differ substantially from the retrieval literature as our intention is not to classify an object, but rather to assess similarity in a dictionary space of frames that are also close to each other in physical space. In the following sections, we will describe the different elements of the pipeline, including a discussion of how we extended the gradient fields of SIFT to support multi-directional Gabor filtering, with a corresponding descriptor.

4.1. Preprocessing

The incoming frames for both the database creation and query branch are first converted to greyscale. The images are then downsampled to size 208×117 pixels, which we found sufficient to generate reasonable localisation. Prior to the feature extraction stage, the images were pre-smoothed at a scale of $\sigma = 1.2$, which corresponds to VLFEAT's default keypoint scale $s = 2$. This avoids computing a Gaussian scale space: the single scale of descriptor calculation on a dense grid in which a single value of σ appeared well-suited to the goal of working with relatively small images. More information on these design choices, and a comparison of sparse versus Dense SIFT, and other descriptors, is discussed elsewhere [56].

4.2. Patch descriptors

We used two different types of patch descriptors in our studies of appearance-based localisation. One of these was optimised for

speed in the client–server application, and the other was optimised for accuracy of localisation. Both were used in the same BoVW pipeline. The DSIFT descriptor [38,41] was selected for its wide availability within many (operating system and software) environments; we used the implementation within the VLFEAT [70] library. We used a stride length of 3 pixels [56]. This produced around 2, 000 descriptors per video frame, each descriptor representing a patch of roughly 10 × 10 pixels.

We also sought to explore the best possible localisation performance that was obtainable in a reasonable amount of computational time. To do this, we used a slightly longer descriptor based on Gabor filtering, as explained in detail in our previous work [56]. Multi-directional spatial Gabor filters are attractive for visual analysis because, amongst their many uses in computer vision, they can be used to characterise image textures [1,33,76] and perform face recognition [78]. Both of these are applications of computer vision that hold potential for visually impaired users. Although we did not use direct mapping of texture to tactile feedback in this study, it is a feature that we plan to investigate in the future (see, for example, the texture mapping work of [1]).

As spatial filtering is computationally expensive, we do not wish to use both a gradient field descriptor (e.g. HoG or SIFT) *and* texture filtering. It would be sensible to create descriptors that are comparable with SIFT from the outputs of directional Gabor filtering. In that way, the same convolution operation can be re-used for both mapping of image texture to tactile textures and for location recognition. These new descriptors were created by adding pooling and sampling operators to the outputs of a Gabor filter bank in order to construct descriptors that are applicable to BoVW techniques [49]. We now describe the theoretical approach to the creation of the Gabor-based descriptors.

4.3. Descriptors from Gabor filters

Recognising that SIFT operates with vector fields of the form:

$$\vec{\nabla} f(x,y;\sigma) = \frac{\partial f(x,y;\sigma)}{\partial x}\vec{x} + \frac{\partial f(x,y;\sigma)}{\partial y}\vec{y} = \frac{\partial f(i_1,i_2;\sigma)}{\partial i_1}\mathbf{i}_1 + \frac{\partial f(i_1,i_2;\sigma)}{\partial i_2}\mathbf{i}_2 \quad (1)$$

$$= \bigcup_{k=1}^{2} \mathcal{D}_k[f(i_1,i_2;\sigma)]\mathbf{i}_k \quad (2)$$

where spatial dimensions (x, y) are now represented by modes i_1, i_2 in the tensor notation of Kolda [36], and Eq. (2) follows from Eq. (1) because of the orthogonality of unit vectors $\vec{x}$ and $\vec{y}$. $\mathcal{D}_k$ is a derivative operator along dimension (mode) i_k.

More generally, when the directional operators are not necessarily partial derivatives, we may introduce the discrete spatial orientation tensor at scale σ as:

$$\mathsf{G}_\sigma = \bigcup_{k=1}^{K} \mathcal{O}_k[f(i_1,i_2;\sigma)]\mathbf{i}_k \quad (3)$$

The operator $\mathcal{O}_k$ is some form of discrete, directional spatial operator. Eq. (3) generalises a two-dimensional gradient field at scale σ; it permits more than 2 directions of peak angular sensitivity, and unlike the operator $\mathcal{D}_k$, there is no requirement that $\mathcal{O}_k$ be linear.

Using oriented Gabor filters, an order 3 tensor $\mathsf{G}_{\sigma,\lambda}$ is constructed by:

$$\mathsf{G}_{\sigma,\lambda} \triangleq R_+\left(\mathsf{F} \underset{\{(\sim,i_3)\}}{\overset{\{i_1,i_2\}}{[*]}} \mathsf{K}_G\right) \quad (4)$$

where $[*]$ represents *tensor convolution* in the modes i_1 and i_2 (see Appendix B), and λ is a tunable spatial wavelength parameter. K_G is an order 3 tensor of dimension $7 \times 7 \times 8$. The function $R_+(\cdot)$ is the one-sided ramp function applied element-wise to its tensor-valued argument. i.e. for a tensor with elements $a_{i_1,i_2,\ldots,i_N}$ it creates a tensor of the same order and size with the elements $|a_{i_1,i_2,\ldots,i_N}|$. The tensor K_G holds antisymmetric Gabor functions, one direction per slice of the third mode (i_3); directions span the 2D plane.

To create the descriptors from the order 3 tensor $\mathsf{G}_{\sigma,\lambda}$, a *permuted tensor convolution* (Appendix B) is applied between $\mathsf{G}_{\sigma,\lambda}$ and a *pooling tensor* P:

$$\mathsf{D} \triangleq \mathsf{G}_{\sigma,\lambda} \underset{\{i_3\}}{\overset{\{i_1,i_2\}}{[*]}} \mathsf{P} \quad (5)$$

The pooling tensor, also of order 3, defines 17 pooling regions with respect to each location in image space, distributed in a radial and angular fashion across a patch; the values in this tensor are visualised over a normalised neighbourhood of unit width and height in Fig. 4. The resulting order 4 tensor, D, may be reshaped [36] into an order 3 tensor containing 136 slices along mode i_3. Descriptors are obtained by sampling D every 3 pixels along modes i_1 and i_2, generating around 2,000 descriptor vectors per frame, each of 136 elements. This makes the number of descriptors comparable to DSIFT, a design goal that helps in planning storage demands, and in interchanging modules.

Both the pooling patterns and the Gabor parameters σ and λ were optimised on the PASCAL VOC 2007 database [25] for retrieval accuracy. No further optimisation was applied for the experiments on the RSM dataset as described in this paper.

4.4. BoVW pipeline

In order to test the ability to localise position based on the visual structure of either a short sequence of frames or individual frame information, we adopted a retrieval structure for efficient mapping of the visual descriptors, sparsely or densely populating an image, into a single frame or vignette-level [57] representation. The approach is based on standard retrieval architectures used for image categorisation—the Bag-of-Visual-Words (BoVW) model—and is illustrated in Fig. 5.

For the vector quantisation, hard assignment was used to encode each descriptor vector by assignment to a dictionary entry. The dataset was partitioned by selecting $N_v - 1$ of the N_v video sequences of passes through each possible path. This ensured that queries were *never* used to build the vocabulary used for testing the localisation accuracy. The dictionary was created by applying the k-means algorithm on samples from the video database. We fixed the dictionary size to 4000 (clusters, words); this allows comparison with the work of others in related fields [16].

The resulting dictionaries were then used to encode the descriptors: both those in the database and those from queries. The frequency of occurrence of atoms was used to create a histogram of visual words "centred" around each frame of the video sequence (visual path) in a database, and the same process was used to encode each possible query frame from the remaining path. All histograms were L_2-normalised.

4.5. Localisation using "kernelised" histogram distances

Once histograms had been produced for all the images in the database we generated "head-to-head" distance comparisons between each pair of passes. In the image retrieval and object recognition literature [71], we call *kernel matrices* those that contain a "kernelised" version of a form of scalar products between the feature vectors (the histograms) of each element in one class against each element of another class. In other words, that scalar product is in reality a distance metric, and in our case, the similarity of

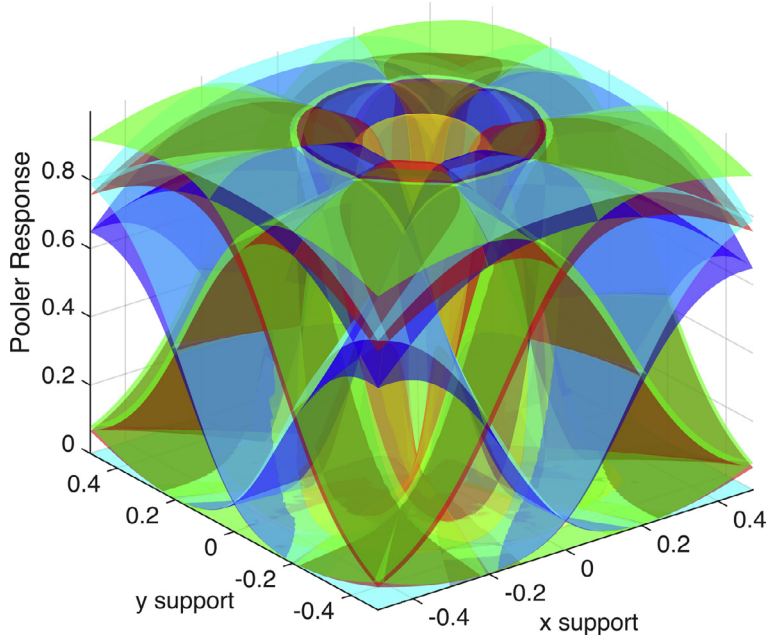


Fig. 4. Patterns of 17 spatial poolers consist of regions in a centre-surround organisation, with angular variation. These pooling weights were optimised on the PASCAL VOC 2007 dataset for categorisation by an optimisation approach. Colours are chosen to alternate in order to allow spatial relationships to be visible; *x*, *y* spatial scales are relative to patch size.

4000-dimensional histograms is performed for each query frame against the database entries.

Using n to denote the frame number, p is a particular journey down a corridor, and q a specific query frame, the χ^2 kernel (Eq. (6))

$$K_{\chi^2}(H_q, H_{p,n}) = 2\frac{(H_q \cdot H_{p,n})}{H_q + H_{p,n}} \tag{6}$$

and the Hellinger kernel (Eq. (7))

$$K_H(H_q, H_{p,n}) = \sqrt{H_q \cdot H_{p,n}} \tag{7}$$

are common choices to compare query frames encoded by a BoVW encoded frame with a database containing several frames (here, consisting of different journeys, p and frames, n). In this work we chose the χ^2 kernel, as it performed better on the task of path localisation [56]. For a random subset of the $N_v - 1$ videos captured over *each* path in the dictionary, the query is selected from amongst the frames of the remaining journey. Each histogram, H_q, representing a query frame results in $N_v - 1$ separate comparison matrices (Fig. 6), each containing the distances of each database frame histogram to the query in the form of matrix columns.

We identified the best matching frame, $\hat{n}$ from pass $\hat{p}$ across all of the $N_v - 1$ vectors:

$$L(\hat{p}, \hat{n}) = \arg\max_{p,n}\{K_{\chi^2}(H_q, H_{p,n})\} \tag{8}$$

$H_{p,\,n}$ denotes the series of normalised histogram encodings, indexed by p drawn from the $N_v - 1$ database passes, and n denotes the frame number within that pass. The estimated "position", L, of a query was that corresponding to the best match given by Eq. (8); this position is always relative to that of another journey along approximately the same route; the accuracy and repeatability of this in associating locations between passes was evaluated using distributions of location error and Area-Under-Curve (AUC) criteria derived from these distributions as we will see in Section 6. The method is illustrated in Fig. 6.

5. A tactile interface for a client–server assistive localisation system

We have described localisation systems that use visual input to provide location information by matching queries against a database of previously acquired images of the environment. We now describe how this information can be conveyed to blind and partially sighted users by means of a haptic interface. In Section 6.3 we describe experiments to gauge the quality of the haptic feedback for localisation.

5.1. The Senseg™ App

The goal of the Senseg™ App was to convey localisation information to visually impaired users. The Senseg™ device allows different textures to be felt at different locations and at a varying range of intensities, as specified by the programmer. It provides enough variation in textures to create discretely identifiable objects, and hence impart localisation information through haptic feedback.

5.1.1. Overview of App

We defined two important criteria for the App: 1) manual intervention from the user should be minimised 2) the space available for feedback should be maximised. To address the first criterion, the App was programmed to take photos automatically at fixed intervals. This removed the need for any buttons, allowing the map to be scaled to fit the 7 inch screen of the Senseg™ device. Fig. 7 shows a screenshot from the App, with colour-coded information to provide additional visual feedback:

1. The yellow outline represents walls—the limits of the map—and imparts the greatest intensity of haptic feedback.
2. The grey lines form a grid system. A grid system was used for two main reasons:
   (a) There need to exist distinct boundaries between haptic feedback positions to allow the user to differentiate between them.

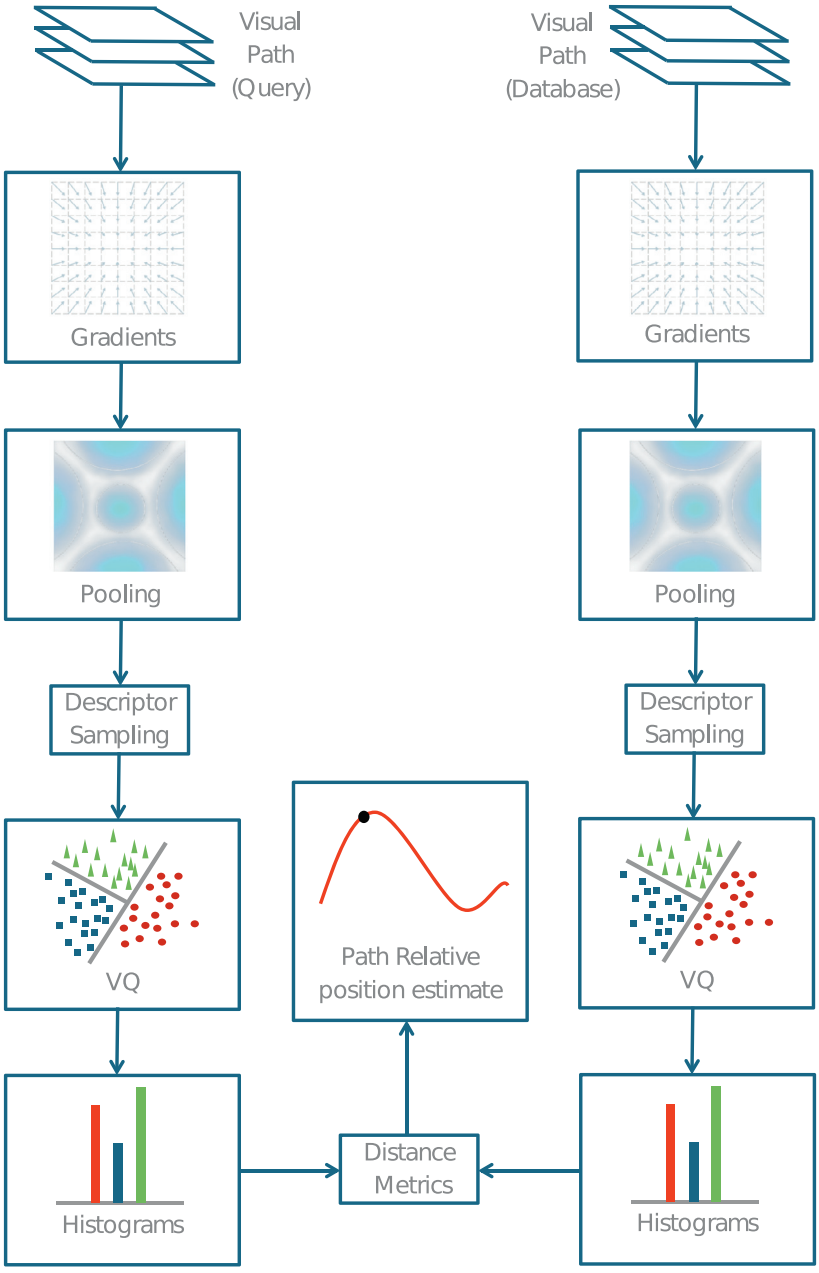


Fig. 5. Video sequences from wearable and hand-held cameras are processed using a customised BoVW pipeline. The pipeline illustrates the idea of comparing BoVW word encodings from individual query frames against a collection of frames from previous journeys.

(b) They allow the user to quantify how far they are from reference points. For example, here the map consists of 10 boxes between the entrance and exit. Each box, therefore, represents 10% of distance between the start and the end of the corridor. This allows the user to estimate their current location by using relative distance between the start and end points.

The perimeter of the boxes have the "Edge Tick" haptic feedback assigned to them.

3. The green box represents the user's estimated position at any given time. The whole area of the box has a "Grainy" texture assigned to it. This allows users to identify their location along their journey.
4. The red box represents the location of the user's touch on the screen over any of the boxes in the grid. This was used to ensure that the App was registering touches correctly during experiments.

Distances are measured in a normalised scale from 0 to 1—this was used as a journey-relative distance metric (in our case between the beginning and the end of the corridor). The normalised scale allows ready adaptation to different tactile screen geometries and methods of user-interaction. As a measure of distance travelled along a desired path, it also easily conveys a sense of how fast one is making progress along a planned route. The percentage bar at the bottom of the App provides fine-grained distance information for testing purposes.

5.1.2. Task flow

The task flow of the user is to obtain location information as they progress along their journey. To accomplish this we integrated the App with a server that takes images as input and outputs location information. It would be inconvenient for visually impaired users to manually request localisation information, so a picture of

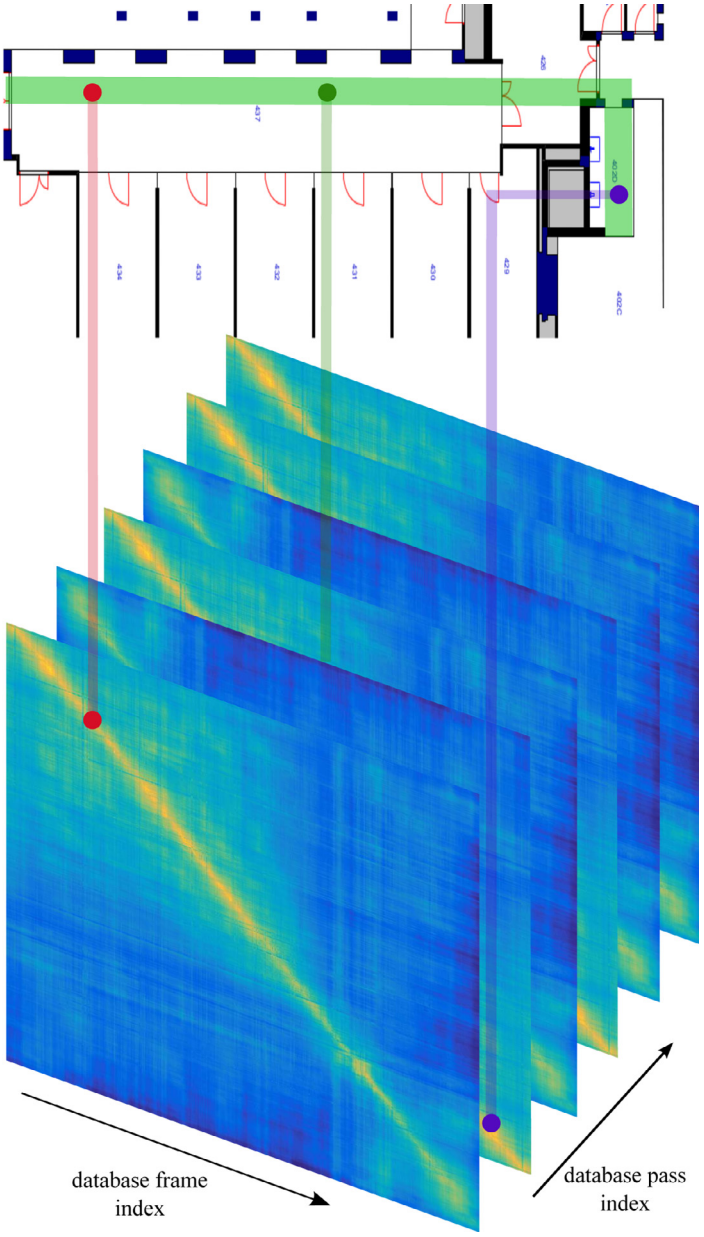


Fig. 6. Matching locations by selecting maximum similarity kernel score between query and database frames. The scores may be obtained by comparing a BoVW encoding of a current query frame against all previous frames acquired from different journeys having similar start and end points. Because the frames are relatively small, comparisons and descriptor calculation for all frames can be rapid.

the user's surroundings is taken automatically at fixed time intervals. This is then uploaded to the server which returns the normalised journey distance.

5.2. Client–server integration

Although the device running the App has a camera, the user could alternatively utilise a wearable camera, such as *Google Glass*, to ease the capturing of visual data. This would be paired with the App on the haptic device. As the user navigates the environment, the camera takes low resolution pictures at regular intervals. The intervals can be chosen to minimise processing/battery usage, whilst still providing responses that are usable in real-time. Each picture is sent to the HTTP server via a POST request to a specific URL endpoint. The HTTP server asynchronously saves the image and calls the appearance-based matching code. This code returns the estimated location to the Senseg^{TM} tablet via the HTTP response. Under the assumption that the indoors area has a Wi-Fi network, we have chosen to offload the computation to a server at the cost of bandwidth [64]. This arrangement is supported by the bandwidth requirements of the appearance-based approach that we settled on. This is because, in contrast to SLAM-based algorithms, the appearance-based method appears to work with quite small images, requiring no more than ≈40 kB per greyscale image, and no more than 120 kB per colour image.

6. Experiments

6.1. RSM dataset acquisition

A total of 60 videos was acquired from six corridors of a large building. A LG Google Nexus 4 phone and a Google Glass (Explorer Edition) were used to capture videos. The Senseg^{TM} tablet was not used in the creation of the dataset and was therefore only used

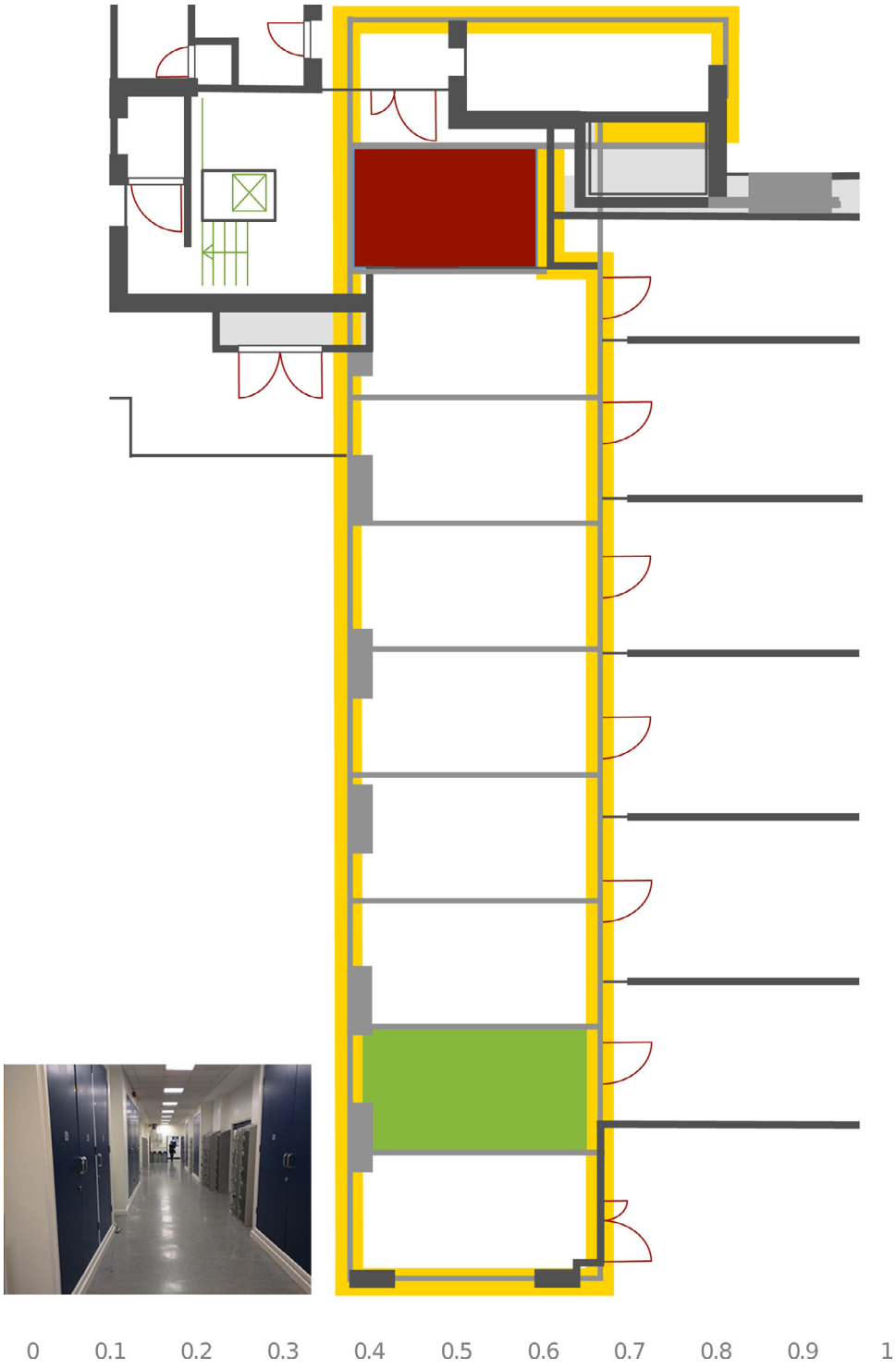


Fig. 7. Senseg™ App screen. The yellow outline represents walls. The grey lines form a grid system for relative localisation. The green box identifies the user's estimated location. The red box depicts the location of the user's touch, and was used for debugging purposes. The horizontal scale at the bottom indicates relative position in the journey. The camera image is also displayed for debugging purposes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

for capturing images for querying location. The main part of the database contains just over 3.0 km of indoor navigation, all with ground truth. For each corridor, ten passes (i.e. 10 separate walks) were obtained. The Nexus 4 (running Android 4.4.2) acquired data at approximately 24–30 fps at two different resolutions, 1280 × 720 and 1920 × 1080 pixels. The Google Glass acquired data at 30 fps with a resolution of 1280 × 720 pixels. The length of the sequences varied from corridor to corridor. Acquisitions were performed at different times of the day and night, so that the lighting varied quite significantly even in the same corridor. Activities (such as cleaning or shifting of furniture) were unplanned, but are to be seen in some sequences. Occasionally, people appear in the videos.

A surveyor's wheel, with a precision of 10 cm and error of ± 5% cm was used to record distances. This was wired to a Raspberry Pi,

synchronised to network time. Timestamp data from the video was used to align ground truth measurements with frames. The dataset is publicly available for download at http://rsm.bicv.org [55].

6.2. Experiments on localisation: appearance-based methods versus SLAM

We compared the best performing appearance-based method against the current state-of-the-art in SLAM for indoor sequences, LSD-SLAM. We first ran the LSD-SLAM code over the 60 video sequences of the RSM dataset and retrieved the results from the visual odometry engine in order to obtain position estimates for each processed frame. This provided an estimate of the distance travelled by a user, allowing comparison to the results we had for the appearance-based method.

The cameras in our devices are quite different to those suggested for use with LSD-SLAM (recommended: monochrome global-shutter camera with fish eye lens); therefore, we had to modify the standard LSD-SLAM software to recover from lost tracking when this happened. We adapted the semi-dense SLAM parameters for our conditions and data, specifically, to minimise instances of lost tracking in medium resolution versions of the RSM dataset, i.e. at a considerably higher resolution than the image required for the appearance-based approach. In Appendix D we provide the parameter values that we used for the experiments.

6.2.1. Measurements of performance

In order to quantify the accuracy of estimating position along physical routes taken by a person with a hand-held or wearable camera within corridors, we selected $N_v - 1$ passes (complete videos) along single corridors contained; these were placed into the journey database. We then randomly selected query *images* from the remaining pass. Using the ground truth, we were then able to get an estimate of the error in localisation. The same principle may be applied to LSD-SLAM; this allows us to test the reproducibility of journeys along the same corridors when using different cameras, and by using different inference techniques. We also summarised localisation errors in the form of an estimate of $P(|\epsilon| < x)$, which is explained in more detail in the following sections.

6.2.2. Cumulative distribution functions

Estimating cumulative distribution functions (CDFs) over absolute positional error allowed us to compare the error distributions of several appearance-based techniques [56]. We also assessed the variability in error distributions when 1 million permuted queries were performed by cycling through 1000 permutations of 1000 randomly selected queries. This Monte-Carlo approach allowed us to get an impression of the stability of the appearance-based matching. All the results were generated with videos resized down to 208 × 117 pixels; these are also supplied with the dataset. We used this metric to compare the performance of the best appearance-based method (the Gabor filter-based descriptor, [56]) and LSD-SLAM. In Section 7 we describe the main findings of this comparison.

6.3. Blindfolded users with tactile sensing

6.3.1. Aim

The aim of this experiment was to evaluate the quality of the tactile feedback when used with blindfolded users who were attempting to estimate their locations. Blindfolded users received tactile cues on the tablet that encoded an estimate of their position along a specific journey relative to the start and end points. Given several location estimates conveyed through the Senseg™ tactile interface, this series of experiments assesses the accuracy of tactile feedback for localisation through a user's perception of his or her position.

6.3.2. Experimental protocol

Eighteen volunteers were asked to conduct the following steps:

1. Firstly, the volunteer was asked to ground his/her hands.
2. The volunteer was then given some familiarisation tasks with a Senseg™ demonstration App that shipped with the tablet ("HapticGuidelines"). This App allows a user to gain familiarity with the feel of the different textures provided within the localisation App. Volunteers were asked to determine which of their fingers appeared the most sensitive to the haptic effects. They were also asked to find the correct finger movement speed to obtain the most feedback.
3. After the nature of the experiment was announced to each volunteer, they were again given the haptic tablet with the localisation App already launched. Two red rectangles denoted the start and end points and had specific—relatively intense—textures. Each of these boxes had a screen size of 150 × 75 "tixels", equivalent to the same number of pixels in the touchscreen display.[1] According to the Nexus 7 (2012) specifications, the screen resolution is 216 ppi (85 ppcm), and so the actual height of each box corresponds to around 0.35 in (0.88 cm).
   The volunteer was then asked to search for four landmarks:
   (a) the beginning of the path,
   (b) an area with no haptic feedback, this was the area that would represent the path that users had already traversed,
   (c) an area with haptic feedback, that represents the remaining segment of the path. This feedback would be the same "Edge Tick" texture described in Section 5.
   (d) the end of the path, with highlighted haptic texture.
4. The volunteer was then carefully blindfolded with a clean tissue being placed between the blindfold and their eyes. The experiment then began.
5. Participants were given 20 tactile cues, each spaced 15 s apart. In the time between the cues, they were asked to estimate and announce their location estimate to the closest 10%; 0% was the starting point of the journey and 100% was the end point of the journey.
6. After the first 100 trials (five users), it was found that participants were finding it hard to distinguish their whereabouts. This was found to be due to a build-up of static charge on the surface of the screen. From then on, (for the next 13 users), the screen was discharged after every two tactile cues.

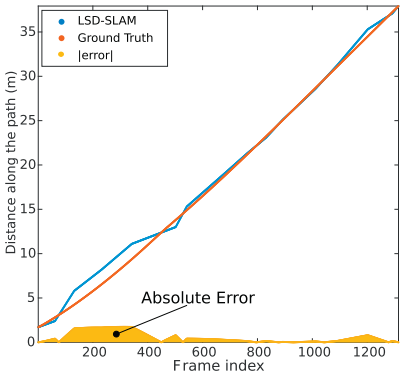
In the following section, the results of the experiments—described above—are presented and discussed. Since one aim of this work has been to compare potential sources of error, we report the experimental outcomes of visual position inference and of ability to convey the inferred position separately, synthesising the implications towards the end of the section.

7. Results

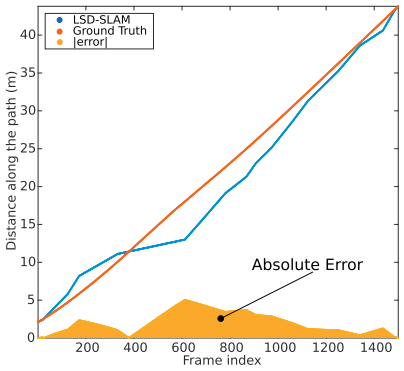
7.1. Performance of vision algorithms

In this section we compare the performance of an appearance-based method with a current state-of-the-art technique in indoor localisation: LSD-SLAM. This section contains comparisons of reproducibility in location sensing within the same corridor on repeat journeys, comparisons of LSD-SLAM error performance within different corridors, and comparisons of LSD-SLAM and appearance-based methods through estimates of the distributions of localisa-

[1] In reality, the App design uses the Android "Supporting Multiple Screens" API from the software development kit [20], which establishes a density-independent pixel (dp) for its responsive and multi resolution design. For the case of our Nexus 7 (2012) tablet, with a resolution of 216 ppi (85 ppcm), we used 100 × 50 dp, which converts to 150 × 75 pixels.



(a) Errors obtained from one experiment using LSD-SLAM. The maximum absolute error was just over 1 m.



(b) Errors obtained from another experiment using LSD-SLAM. In this case, the absolute error peaked at around 5 m.

Fig. 8. Localisation performance in LSD-SLAM; this shows that in different corridors, the accuracy of LSD-SLAM can change quite significantly. See the text for details of the SLAM parameters and the nature of the dataset, but note that these were obtained at an image resolution of 1024 × 576 pixels. At lower resolutions, loss of tracking dominated the experiments. The difference in the x and y axis labelling is because experiments in a) and b) are obtained from two different corridors with different lengths and different numbers of frames. (For interpretation of the references to colour in the text, the reader is referred to the web version of this article.)

tion error over different distances; the last of these are also summarised through AUC metrics.

7.1.1. Performance of LSD-SLAM

Fig. 8 (a) and (b) illustrate the localisation performance of LSD-SLAM (blue) with respect to the ground truth (red). The two cases that we show are selected to illustrate that accuracy of localisation can vary, depending on the specific corridor within the RSM dataset (see Section 6.1). As can be seen from the figures, in the better case, the absolute error is below 2 m. However, in some cases, we found errors as high as 5 m. These errors were found when operating at image sizes that allowed LSD-SLAM to function without losing tracking.

7.1.2. CDF comparisons

In Appendix C, we describe the algorithm to generate the cumulative distribution functions (CDFs) of localisation error. CDFs of error allow us to characterise the distribution of the error in localisation, and could help identify the sources of error. From the CDF, we can also immediately estimate the probability of localisation error being less than x m. This is a quantitative measure that can be used during parameter optimisation, and as a bound on error performance. Secondly, the Area Under the Curve (AUC) of the CDF can be used as a performance measure.

As we can see from Table 1, the appearance-based method using Gabor-based descriptors performed better than LSD-SLAM in the RSM dataset. This does not imply that the technique that we used *replaces* SLAM or its equivalents—but it is a different context of usage. A user who is seeking to get from A to B may be more interested to know that they are passing distinctive visual locations than in mapping out route geometry. In the scenario we describe, and in which we are comparing performance, the journey of a user is assessed against those made by other people who have made the same journey—location becomes journey-relative, not map-relative. Such a usage scenario also means that loop closure may not be possible. A final point to remember is that the cameras that are used in capturing the journeys are not necessarily identical, and certainly may be uncalibrated.

Box and whisker plots that evaluate the reproducibility in localisation are presented in Fig. 9. These illustrate the reproducibility within RSM corridors 1 and 3 (C1 and C3) for multiple "leave-one-out" passes. The plots suggest that whilst LSD-SLAM yields worse results in terms of error, it has a consistency in performance that is comparable to that of the appearance-based method. Also,

Table 1
Cumulative distribution function values against localisation error in metres (ϵ). $P(|\epsilon| < x)$, expressed as a percentage. From this table, the best appearance method achieves a probability of 90% of localising with an error below 2 m, whilst LSD-SLAM achieves just above a 50% accuracy level for that error boundary, and required larger images. In addition, the performance of LSD-SLAM was significantly worse (compare the columns of minimum performance) on some corridors and journeys.

	Appearance-based			SLAM		
$\lvert\epsilon\rvert$ (m)	Min(P)	Mean(P)	Max(P)	Min(P)	Mean(P)	Max(P)
0.25	22.18	23.50	25.18	5.21	5.92	6.98
0.50	51.70	53.45	55.35	14.08	15.14	16.26
0.75	65.78	67.31	69.08	21.35	22.75	24.11
1.00	71.25	72.73	74.28	28.11	29.40	30.69
1.25	78.98	80.28	81.96	34.31	35.64	36.98
1.50	83.76	85.10	86.31	39.85	41.26	42.62
1.75	88.36	89.43	90.40	45.54	46.92	48.51
2.00	92.13	92.99	93.80	49.88	51.32	52.70
2.25	93.46	94.23	94.86	53.77	55.27	56.73
2.50	95.36	95.95	96.61	58.33	59.88	61.34
2.75	96.50	97.03	97.56	64.09	65.64	67.30
3.00	97.84	98.25	98.59	67.84	69.59	71.04

the errors for LSD-SLAM rarely go beyond 5 m, with an average of $\mu_e = 2.48 \pm 2.37$ m. Conversely, the appearance-based method contains some outliers; even so, for some sequences the error is of the order or lower ($\mu_e = 1.31 \pm 0.39$ m) than the best reported for SLAM.

7.1.3. Area-under-curve comparisons

Using the ROC curves that can be constructed from the cumulative errors shown in Table 1, the AUC for the best appearance-based method ranged from 96.11% to 96.39%. For the case of LSD-SLAM, the AUC ranged from 89.71% to 90.61%. All queries were again performed by adopting the leave-one-journey-out strategy, but because of the high repeatability of results, we did not apply random frame-level sampling in the estimation of these performance measures.

7.2. Blindfolded tactile experiments

Two remarks are in order regarding the haptic tablet device. First, the accuracy of the location feedback improved after discharging the tablet's screen with an electrostatic cloth at regular intervals. Fig. 10a and b show the results for individual subjects;

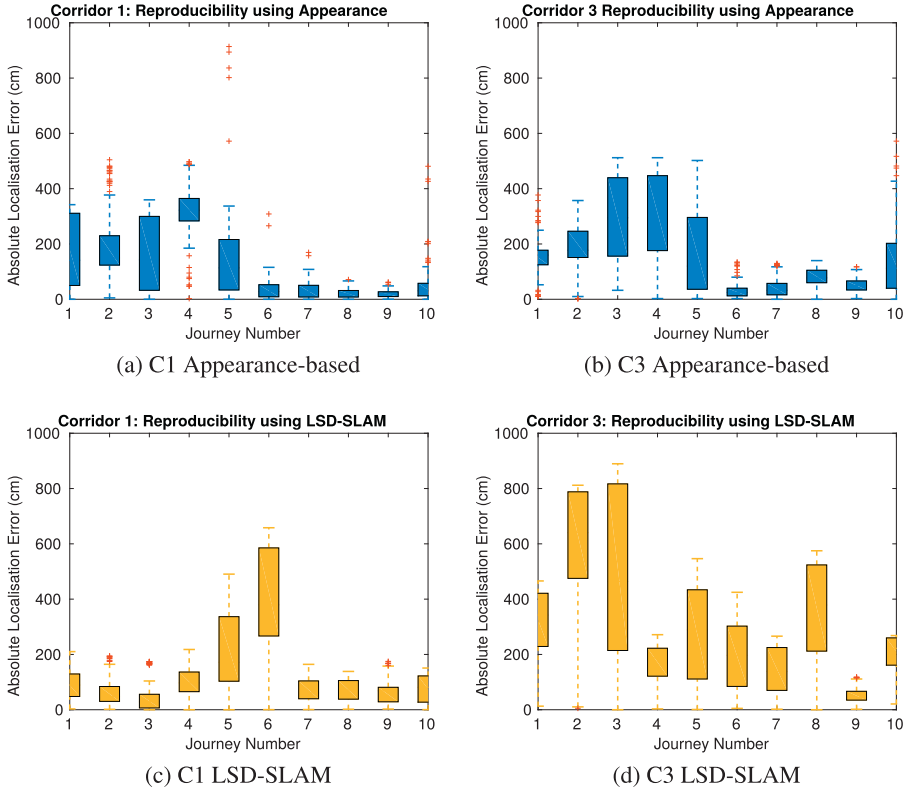


Fig. 9. Box-and-whisker plots depicting the errors obtained in two corridors, using either LSD-SLAM or appearance-based matching. The top row corresponds to the appearance-based result. The bottom row corresponds to LSD-SLAM. On each graph, the horizontal positions correspond to different journeys down the same corridor position-referenced against the remainder of journeys in the database. Each bar represents the statistics of 100 random image queries for that query. These graphs suggest that LSD-SLAM and an appearance-based approach are comparable in terms of reproducibility of localisation within the same corridor. Note, however, that much lower spatial resolution (less than 1/4 of the image size, in pixels) is used for the appearance-based technique than for LSD-SLAM.

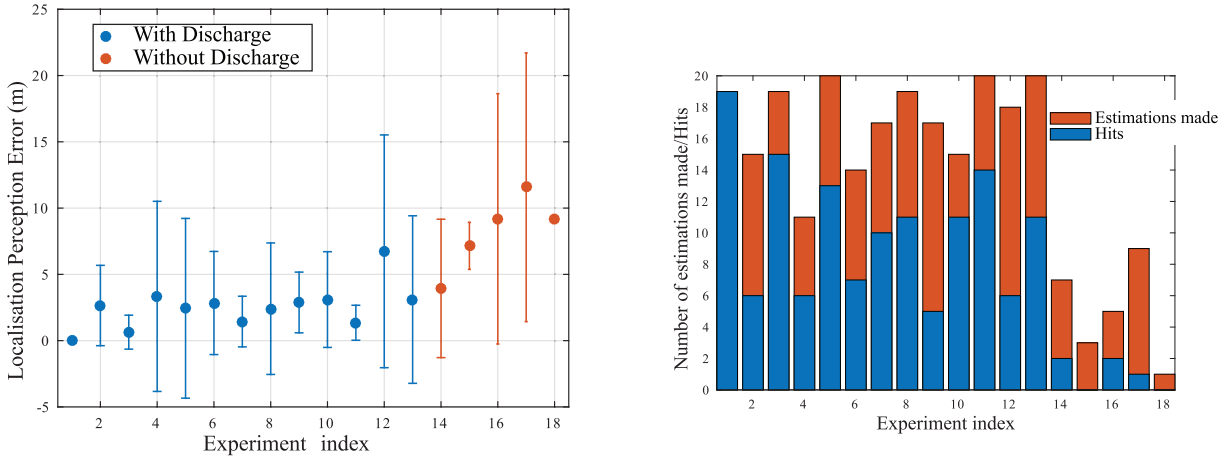


Fig. 10. (a) Errors in localisation using tactile feedback via the Senseg™ tablet. Results from individual subjects spaced along the horizontal axis. (b) Proportion of hits (correct estimates of the portion of the journey completed) together with the number of estimates provided by each subject.

one can see a notable improvement in the users' performance when the device was discharged between trials (summarised in Table 2). Secondly, with our device, haptic feedback could only be discerned when the tablet was plugged into a USB charging port, i.e. when it was grounded. This currently limits its use as a portable device for providing haptic feedback.

Allowing for these limitations, we can see from Table 2 that there is a 58.18% hit rate and an average error of roughly 4 m. For comparison, the grid size we used for a representative test corridor of 30.62 m long with discretised locations 10% apart is $\approx$ 3 m. From Table 1 we can see that, using this size of tactile box, the appearance-based method would give a correct estimate with a mean probability of 98.25%, whilst SLAM would achieve 69.59%.

During the experiment, the user placed their finger at either the end or the beginning blocks, which were a different texture ("Grainy") to the map position ("Bumpy"). The position was updated on the map every 15 s, providing ample time for confident estimates. However, most users found it helpful to count blocks (edge tactile feedback) from the current position to the end block through the tactile feedback between these points.

Another observation is the time participants took to estimate their location: users took an average of 11.28 s ($\sigma = 5.58$ s) to

Table 2
Summary of the results of tactile feedback experiment. A precision metric can be calculated as $prec = \frac{hits}{estimates}$.

Discharge	MeanErr	StdErr	No. of trials	No. of estimates	No. of hits	Precision
Yes	2.63 (m)	4.04 (m)	260	231	136	58.87 (%)
No	9.28 (m)	5.34 (m)	100	44	24	54.54 (%)
Overall	4.11 (m)	4.33 (m)	360	275	160	58.18 (%)

complete the task. This might seem a long time in a real navigation scenario. However, this experiment required the user to provide estimations on randomly selected locations, without any correlation between the tactile cues. Another factor is that the experimental protocol suggested a 15 s time for participants to provide estimates of their location, which might seem long. During a typical real journey which has been planned using the map, there would be a predictable progression to the order of the cues once a journey has started. Only if a trajectory change were to happen would a user need to spend significant time finding their position on the map.

A final comment is concerned with the large difference between the errors from the vision system and the errors in the user interface. The vision subsystem had an average absolute error of 2.43% in the sequences of the tactile experiment. This represents sub-metre accuracy of 0.75 m for a journey of 30 m in length. The error of the haptic feedback is significantly larger than the error of the live system's algorithm. This suggests that an active spatial "zooming" of the map needs to be included in order to reduce location error from tactile feedback.

8. Conclusion

In this paper, we have described a prototype indoor visual localisation system which provides location information on a tactile map. The proposed system is able to exploit data crowdsourced from consumer-grade devices in order to build up its database. We have also explored the use of image data from hand-held or wearable cameras to pinpoint a user's position relative to other journeys made along the same physical route. Information on the layout of a route was provided to users through haptic cues via a Senseg^{TM} tablet (a Google Nexus 7 device modified to allow extra haptic feedback). We found that the architecture of such a system can be remarkably simple. We have also described the components of our prototype: the algorithms for computer vision, the architectural aspects of the App, and the client–server interaction.

On comparing the performance of an appearance-based method of localisation to one of the more recent SLAM algorithms, LSD-SLAM, we were surprised to find that appearance-based localisation was at least as accurate as that of LSD-SLAM without loop closure over distances of around 50 m. We also found that appearance-based localisation was achievable with low resolution images (208 $\times$ 117 pixels), as compared to LSD-SLAM (minimum size 1024 $\times$ 576 pixels). This lowers the computational burden, a potentially important factor in an assistive context, where the power autonomy of devices is an important requirement. In addition, the modest image size requirements for appearance-based localisation reduce the bandwidth and storage needed for crowdsourcing data. We found that 1500-frame sequences, sufficient for a 50 m corridor at normal walking speeds ($\sim$1.4 m/s), consumed no more than 2 MB once compressed, meaning that the journey segments required for localisation can feasibly be harvested from several users who have made the same journey.

We evaluated the accuracy of blindfolded volunteers to perceive location via haptic cues presented on a map. This map was laid out on a tactile display, with haptic cues to indicate boundaries and also start and end points. We found that blindfolded users were able to perceive their location on a map with a minimal amount of practice. This supports the hypothesis that the use of visual indicators of map-relative position can be replicated by haptic technology. Specifically, this approach has the potential to allow a spatial layout and a user's position to be conveyed via haptic feedback, an approach that may be added to smartphone-based technologies that can be used by visually-impaired users to navigate indoors.

Taking both the precision of the haptic device and the accuracy of visual localisation into account, we suggested a technique to infer the error in localisation that reflects the limits of the haptic device and the position sensing technology. In effect, this allows us to determine the error in localisation that a given grid size on the tactile map may have for a specific journey. Whilst no technology system for localisation is without errors, using the CDF as suggested in Section 7 allows sensible measures of accuracy to be cited: this represents a step towards being able to characterise the performance of devices with haptic feedback for navigation in a more reproducible manner.

In future work, we plan to extend two aspects of the work reported in this paper: improving the visual processing and further exploring the capabilities of mapping visual information onto haptic devices. Generally speaking, the use of images captured with wearable cameras by a navigating person remains only superficially explored in the literature. Though power consumption and accuracy of detection remain key barriers to wide scale deployment, these barriers will be lessened over time. In addition to location estimation, the possibility of using cameras to detect obstructions, people, and any deviations in the environment from previous journeys, holds promise for richer assistive technology for navigation. We also plan to integrate ground-plane detection using a wearable camera in order to find irregularities in walking surfaces, or obstructions out to around a 5 m distance. Finally, other sources of data, for example Wi-Fi signal strength, can play a key role in both improving the reliability of position estimation, and overall robustness for cases such as failure of indoor lighting.

On the haptic side, we plan to refine the mapping from floor plans to tactile feedback. Returning to the Senseg^{TM} platform as an example, a variety of textures could be conveyed to a user by varying the amplitude and temporal pattern of voltage pulses sent to the haptic interface. By combining the flexibility of this device with prior work on mapping textures to haptic feedback, it should be possible to improve the information conveyed to a visually impaired user by automatically translating visual information from an appropriately prepared map. In addition, a standard map format that contains hatches or textures to illustrate locations of steps, or different types of rooms, could be mapped to different tactile sensations on devices with the appropriate range of haptic feedback. The wider implications for mapping visual structure to tactile sensations extend—via the use of the same Gabor functions suggested for appearance-based localisation—to features in the live video feeds from a wearable camera, opening new possibilities for conveying visual information into haptic cues for navigation.

Acknowledgments

This work was partly supported by the UK's Engineering and Physical Sciences Research Council.

Table A.3
List of symbols.

Symbol	Meaning
A, B	General, arbitrary tensors of orders N_A and N_B used in definitions
F	Order 2 tensor denoting an intensity image
K_G	3*rd* order Gabor kernel tensor
$K_\cdot$	where $\cdot$ is either χ^2 or H, denoting "kernelised" distance measures.
i_1, i_2, i_3...	Tensor indices of increasing mode, where each i_n takes integer values from 1 to $I_n, \forall n$
D	Descriptor tensor
$\mathcal{D}_k$	Partial derivative operator in direction k
$H_\cdot$	A histogram constructed from a query image or a series of database images
L	Likelihood function based on histogram comparisons
K	Number of modes along which a generalised directional operator is applied
$\mathcal{M}$	Set containing the modes over which tensor convolution (Appendix B) is performed; when modes match for the two arguments, mode indices are given, else mode tuples are given
N_v	Number of videos in the complete dataset being studied
$\mathcal{O}_k$	A generalised directional spatial operator in direction k
P	Pooling tensor (order 3 tensor for performing spatial pooling)
$\mathcal{P}$	Set containing the modes over which permutations are performed in permuted tensor convolution (Appendix B)
P	Probability
ϵ	Error (used in distribution of errors)
$f(x, y)$	Image as an intensity function of continuous spatial coordinates
n, p	Frame number and journey number, respectively
σ	General spatial scale parameter; either for Gabor function or Gaussian scale-space
$\vec{\nabla}$	The "gradient" operator; maps scalar field to a potential field
$\frac{\partial\{\cdot\}}{x}$	Partial derivative with respect to x
$\vec{x}, \vec{y}$	unit vectors in a Cartesian coordinate system (resp. $\mathbf{i}_1$, $\mathbf{i}_2$) in Kolda's notation.
x	Single spatial position variable.
$\mathsf{A}\overset{\mathcal{M}}{[*]}\mathsf{B}$	Tensor convolution between equal-ordered tensors A and B; (see Appendix B).
$\mathsf{A}\underset{\mathcal{P}}{\overset{\mathcal{M}}{[*]}}\mathsf{B}$	Permuted tensor convolution between A and B; (see Appendix B).

Appendix A. Symbols

Table A.3

Appendix B. Tensor convolution

We have found it useful to adopt the definitions of [36], in which the tensors are interpreted as multidimensional (multiway) arrays. The authors also introduce or formalise operations upon and between tensors. In Kolda and Bader's notation and nomenclature, the meaning of a *tensor* is different to that of classical physics and stress-analysis, in which tensors are mathematical entities that obey strict transformation laws.

In Kolder and Bader's (K&B's) terminology, the *order* of the tensor is the number of dimensional indices required to address it; for example, an order 5 tensor A may have addressable elements a_{i_1,i_2,i_3,i_4,i_5}, with each index varying from 1 to $I_n, n = 1, 2, 3, 4, 5$ in integer steps; note that in contrast with the K&B notation, indices are comma-delimited. Since each element of the tensor can be restricted to be real-valued, we may consider A as lying in $I_1 \times I_2 \times I_3 \times I_4 \times I_5$- dimensional real space. The *mode* of a tensor refers to the tensor elements simultaneously addressed by one of the indices, and is applied to refer to operations that involve, possibly non-exclusively, a particular one of the indices. Definitions of tensor-vector and tensor-matrix products follow [36], with tensor contraction as described in [11] and also [3].

In the following definitions, we will refer to the tensors A, B and C, where $\mathsf{A} \in \mathbb{R}^{\prod_{n=1}^{N_A} I_n}$ is of order N_A, containing elements $a_{i_1,i_2,\ldots,i_{N_A}}$, and $\mathsf{B} \in \mathbb{R}^{\prod_{n=1}^{N_B} J_n}$ is a tensor of order N_B with elements $b_{j_1,j_2,\ldots,j_{N_B}}$, and C is of order N_C.

Definition 1 (Tensor convolution)**.** We denote the tensor convolution operator in modes $\mathcal{M}$ by the following:

$$\mathsf{A}\overset{\mathcal{M}}{[*]}\mathsf{B} : (\mathsf{A}, \mathsf{B}) \longmapsto \mathsf{C}$$

where $\mathcal{M}$ is a set of $|\mathcal{M}|$ tuples representing paired indices of A and B over which the convolution is performed. These indices associate the modes of the tensors being convolved together; if single mode indices, rather than tuples $(\cdot, \cdot)$ are provided, then it is understood that the modes are repeated for the second element of a tuple.

The tensor convolution operator maps equal-order tensors, A and B to a tensor C by the following:

$$\mathsf{A}\overset{\mathcal{M}}{[*]}\mathsf{B} = \sum_{i'_{m_1}} \cdots \sum_{i'_M} a_{i_1,i_2,\ldots,i'_{m_1},\ldots,i'_M,\ldots,i_{N_A}} \times b_{i_1,i_2,\ldots,i_{n_1}-i'_{n_1},\ldots,i_{n_M}-i'_{n_M},\ldots,i_{N_B}} \tag{B.1}$$

where $\mathcal{M}$, takes the form of a set of tuples that associate indices in A with those in B for the convolution:

$$\{(m_1, n_1), (m_2, n_2), \ldots, (m_M, n_M)\}$$

The order of the result, N_C is equal to that of both A and B: $N_C = N_A = N_B$. however, we do not necessarily have to perform $N_A = N_B$-dimensional convolution using this operator: we can perform convolution in only some of the modes, with the modes that participate being indicated by the elements of $\mathcal{M}$. The number of fibres in A and B must be the same for any dimensions that do not participate in the convolution. For the modes that do not participate in the convolution, the size of C along these modes remains the same as for A and B; for the modes that participate in convolution, the size C is greater than that of A and B in the usual way in which discrete convolution expands support.

Definition 2 (Permuted tensor convolution)**.** We define the *permuted* tensor convolution operator in modes $\mathcal{M}$ permuted over the

modes $\mathcal{P}$ as a mapping taking the form:

$$\overset{\mathcal{M}}{[*]}_{\mathcal{P}} : (\mathsf{A}, \mathsf{B}) \longmapsto \mathsf{C}$$

where $\mathcal{M}$ is a set of $|\mathcal{M}|$ tuples representing paired indices of A and B over which the convolution is performed and $\mathcal{P}$ represents the modes of A and B for which permutation is performed.

The permuted tensor convolution operator maps tensor, A, to an equal or higher-order tensor C by the following:

$$\mathsf{A}\overset{\mathcal{M}}{[*]}_{\mathcal{P}}\mathsf{B} = \sum_{i'_{m_1}}, \dots \sum_{i'_M} a_{i_1,i_2,\dots,i'_{m_1},\dots,i'_M,\dots,i_{p_1},i_{p_2},\dots i_{pp},\dots,i_{N_\mathsf{A}}} \times b_{i_1,i_2,\dots,i_{n_1}-i'_{n_1},\dots,i_{n_M}-i'_{n_M},\dots,i_{\pi(q_1|p_1)},\dots,i_{\pi(q_P|p_P)},\dots,i_{N_\mathsf{B}}} \tag{B.2}$$

where $\mathcal{M}$, consists of the tuples:

$$\{(m_1, n_1), (m_2, n_2), \dots, (m_M, n_M)\}$$

and $\mathcal{P}$ by the tuples:

$$\{(p_1, q_1), (p_2, q_2), \dots, (p_P, q_P)\}$$

The permutation operator $\pi(i|j)$ denotes that the indices of the tensor in a particular mode are permuted over the possible values that that mode can take. The order of the result, N_C, will depend on the orders of the tensors A and B, and the modes participating in the operator $\overset{\mathcal{M}}{[*]}_{\mathcal{P}}$, according to:

$$N_\mathsf{C} = \min(N_\mathsf{A}, N_\mathsf{B}) + |\mathcal{P}|$$

Generally, $\mathcal{M} \cap \mathcal{P} = \emptyset$. As for the case of $\mathcal{M}$, if single elements are given for $\mathcal{P}$, it is understood that the second member of the tuple is the same; where no corresponding dimension exists in one argument, the $\sim$ denotes a null mode in the tuple.

This permutation is not across modes, but within the possible values that one mode can take. By way of example, given a definition of an order 2 tensor A of size 2×3

$$\mathsf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \tag{B.3}$$

and an order 3 tensor, B, of size $2 \times 2 \times 2$ where

$$\mathsf{B} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \Bigg| \Bigg| \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \tag{B.4}$$

then the tensor C defined by

$$\mathsf{C} = \mathsf{A} \underset{\{(\sim,i_3)\}}{\overset{\{(i_1,i_1);(i_2,i_2)\}}{[*]}} \mathsf{B} \tag{B.5}$$

is of order $N_\mathsf{C} = 2 + 1 = 3$, and will be of size $3 \times 4 \times 2$:

$$\mathsf{C} = \begin{bmatrix} 1 & 3 & 5 & 3 \\ 5 & 12 & 16 & 9 \\ 4 & 9 & 11 & 6 \end{bmatrix} \Bigg| \Bigg| \begin{bmatrix} 2 & 6 & 10 & 6 \\ 10 & 24 & 32 & 18 \\ 8 & 18 & 22 & 12 \end{bmatrix} \tag{B.6}$$

Appendix C. Algorithm for generating cumulative error distributions

Algorithm 1

Algorithm 1: Calculation of the error distribution.

Inputs :
Database of kernels, $\mathcal{K}_{c,p,l}$
$c = 1, 2, ..., N_c$, `// corridor index`
$p = 1, 2, ..., N_p$ `// pass index`
Number of permutations, P
Number of random queries, Q
Outputs:
Error Distribution, $\mathbf{X}$

```
// Compute localisation error for all possible queries
for c ← 1 to N_c do
    for p ← 1 to N_p do
        // For each query frame in a pass ...
        foreach q : q ∈ P_p do
            // Take the corresponding kernel computed by
            leave-one-out strategy and get closest
            neighbour
            ρ ← getClosestNeighbor(K)
            // Given the ground truth for that query,
            compute the error
            E_{c,p,q} ← computeError(ρ)
        end
    end
end
k ← 1
for i ← 1 to P do
    for j : j ← 1 to Q do
        e_k ← randomSampling(E)
        k ← k + 1
    end
end
// Compute Cumulative Distribution Functions
X ← computeCDF(e_k)
```

Appendix D. LSD-SLAM parameters

Parameter	Definition	Default	Set to
minUserGrad	Minimal absolute image gradient for a pixel to be used at all. Increase if your camera has large image noise, decrease if you have low image-noise and want to also exploit small gradients.	1.96	5
cameraPixelNoise	Image intensity noise used for e.g. tracking weight calculation. Should be set larger than the actual sensor-noise, to also account for noise originating from discretisation / linear interpolation.	16	2.4
KFUsage-Weight(*)	Determines how often keyframes are taken, depending on the overlap to the current keyframe. Larger: more keyframes.	4	10
KFDist-Weight(*)	Determines how often keyframes are taken, depending on the distance to the current keyframe. Larger: more keyframes.	3	10

(*) The values for the keyframe weights, KFUsage-weight and KFDist-weight were increased following the suggestions of LSD-SLAM authors. By increasing these weights the thresholds to take keyframes are lowered, therefore more keyframes are taken, gaining robustness against tracking at the expense of a larger map, more loop closures and slower processing. Although both affect the

amount of keyframes that are selected, KFDistWeight is an indirect weight applied to the distance between frames that has an influence in the keyframe selection threshold. KFUsageWeight on the other hand, directly modifies the keyframe selection threshold.

Appendix E. Tactile feedback experiment protocol

E.1. Context

The "Visual localisation with tactile feedback" project aims to evaluate the quality of the tactile feedback given by the Senseg tablet in an indoor localisation for the visually impaired context. When navigating a physical path, the user receives a tactile cue that encodes an estimate of their position along that specific path, relative to start and end point. Given several location estimate feedback cues through the Senseg tactile interface, the goal of this experiment is to evaluate how accurate this tactile feedback is based on the user perception of the position they are.

E.2. Experiment protocol

1. The user will be given some familiarisation tasks with the Senseg demo that shipped with the tablet as Android applications. These will be:
   - Familiarise with the different textures with the app "Haptic Guidelines".
2. The user will be given the following instructions:
   You have agreed to take part in the "Visual localisation with tactile feedback" project experiment on tactile feedback quality. The experiment consists of the following tasks:
   (a) You will be given the Senseg tablet you used previously to get familiar with its tactile interface.
   (b) If you visually inspect the path, you will notice two red rectangles that denote starting and end point. These are texture highlighted. As you feel the screen with your finger and move it over the path you will notice four haptic "landmarks" or "events" that can be differentiated:
      i. the beginning of the path,
      ii. an area with no haptic feedback, this is the area that would represent the area that users have already traversed,
      iii. an area with haptic feedback, that represents the remaining segment of the path,
      iv the end of the path, with highlighted haptic texture as event (i).
   (c) You will receive one tactile cue every 15 s, making up to 20 cues.
   (d) Upon the reception of the cue, it will be your task to announce an estimate of your location as a percentage of the total distance. You will only provide estimates that are 10% apart:
      - 0%: starting point of the journey,
      - 10%
      - 20%,
      - ...
      - 80%,
      - 90%
      - 100%: end point of the journey.
   (e) As agreed, you will blindfold yourself for this experiment. Please, proceed to wear the blindfold now, the experiment will start shortly.
3. The experiment will start:
   (a) The user will receive 20 tactile cues corresponding to 20 randomised location estimates provided by the localisation server.
   (b) The users' announced estimates will be annotated next to their corresponding index in the following table[2]:

Trial index	True location	Estimated location
1	0.8147	
2	0.9058	
3	0.1270	
4	0.9134	
5	0.6324	
6	0.0975	
7	0.2785	
8	0.5469	
9	0.9579	
10	0.1576	
11	0.4854	
12	0.8003	
13	0.1419	
14	0.4218	
15	0.7922	
16	0.6557	
17	0.0357	
18	0.9340	
19	0.3968	
20	0.2672	

Important note: The apparently high precision on location (second column) allows us to extend the system to measure accuracy that is relevant to pedestrian navigation, but over journeys containing much larger spatial scales.

E.3. Informed consent form

E.3.1. Experiment purpose and procedure

The purpose of this experiment is to evaluate the quality of the tactile feedback given by the Senseg tablet in an indoor localisation for the visually impaired context.

The experiment consists of two parts as detailed in the previous section.

After the experiment, you will be asked to complete a feedback form.

Please note that none of the tasks is a test of your personal intelligence or ability. The objective is to test the usability of our research systems.

E.3.2. Confidentiality

The following data will be recorded: estimates of the tactile-encoded position along a path based on Senseg haptic feedback.

All data will be coded so that your anonymity will be protected in any research papers and presentations that result from this work.

E.3.3. Finding out about result

If interested, you can find out the result of the study by contacting the researcher Jose Rivera-Rubio, after 1 April 2015.

His email address is jose.rivera@imperial.ac.uk.

E.3.4. Record of consent

Your signature below indicates that you have understood the information about the "Tactile feedback with Senseg" experiment and consent to your participation. The participation is voluntary and you may refuse to answer certain questions on the questionnaire and withdraw from the study at any time with no penalty. This does not waive your legal rights. You should have received a copy of the consent form for your own record. If you have further questions related to this research, please contact the researcher.

[2] Not present in the volunteer's copy.

References

[1] W. Adi, S. Sulaiman, Texture classification using wavelet extraction: an approach to haptic texture searching, in: Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), IEEE, 2009, pp. 434–439.

[2] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S.M. Seitz, R. Szeliski, Building Rome in a day, Commun. ACM 54 (10) (2011) 105–112.

[3] S. Aja-Fernández, R. de Luis García, D. Tao, X. Li, Tensors in Image Processing and Computer Vision, Springer Science & Business Media, 2009.

[4] M. Akamatsu, I.S. MacKenzie, Movement characteristics using a mouse with tactile and force feedback, Int. J. Hum. Comput. Stud. 45 (4) (1996) 483–493.

[5] A. Aladren, G. Lopez-Nicolas, L. Puig, J.J. Guerrero, Navigation assistance for the visually impaired using RGB-D sensor with range expansion, IEEE Syst. J. PP (99) (2014) 1–11.

[6] P.F. Alcantarilla, L.M. Bergasa, F. Dellaert, Visual odometry priors for robust EKF-SLAM, in: IEEE International Conference on Robotics and Automation (ICRA), 2010, pp. 3501–3506.

[7] P.F. Alcantarilla, S.M. Oh, G.L. Mariottini, L.M. Bergasa, F. Dellaert, Learning visibility of landmarks for vision-based localization, in: IEEE International Conference on Robotics and Automation (ICRA), 2010, pp. 4881–4888.

[8] P.F. Alcantarilla, J.J. Yebes, J. Almazán, L.M. Bergasa, On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2012, pp. 1290–1297.

[9] A.M. Ali, M. Nordin, Indoor navigation to support the blind person using true pathway within the map, J. Comput. Sci. 6 (7) (2010) 740.

[10] N. Asamura, N. Yokoyama, H. Shinoda, Selectively stimulating skin receptors for tactile display, IEEE Comput. Graphics Appl. 18 (6) (1998) 32–37.

[11] B.W. Bader, T.G. Kolda, Algorithm 862: MATLAB tensor classes for fast algorithm prototyping, ACM Trans. Math. Softw. (TOMS) 32 (4) (2006) 635–653.

[12] D. Barth, The Bright Side of Sitting in Traffic: Crowdsourcing Road Congestion Data, http://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html, 2009.

[13] N. Bhushan, C. Lott, P. Black, R. Attar, Y.-C. Jou, M. Fan, D. Ghosh, J. Au, CDMA2000 1 × EV-DO revision a: a physical layer and MAC layer overview, IEEE Commun. Mag. 44 (2) (2006) 37–49.

[14] J.C. Bliss, M.H. Katcher, C.H. Rogers, R.P. Shepard, Optical-to-tactile image conversion for the blind, IEEE Trans. Man Mach. Syst. 11 (1) (1970) 58–65.

[15] A. Chang, C. O'Sullivan, Audio-haptic feedback in mobile phones, in: CHI'05 Extended Abstracts on Human Factors in Computing Systems, ACM, 2005, pp. 1264–1267.

[16] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: Proceedings of the British Machine Vision Conference, vol. 2, 2011, p. 8.

[17] A. Chekhchoukh, N. Vuillerme, N. Glade, Vision substitution and moving objects tracking in 2 and 3 dimensions via vectorial electro-stimulation of the tongue, in: ASSISTH'2011: 2ème Conférence Internationale sur l'accessibilité et les systèmes de suppléance aux personnes en situations de handicaps, "De l'usage des STIC à une plus grande autonomie: des recherches interdisciplinaires", HAL Open Archives (France), 2011, p. 16.

[18] M. Cummins, P. Newman, Appearance-only SLAM at large scale with FAB-MAP 2.0, Int. J. Robotics Res. 30 (9) (2011) 1100–1123.

[19] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: real-time single camera SLAM, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 29 (6) (2007) 1052–1067.

[20] G.A.A. Documentation, Supporting Multiple Screens, 2015.

[21] W. Dong, D. Olguin-Olguin, B. Waber, T. Kim, A. Pentland, Mapping organizational dynamics with body sensor networks, in: IEEE International Conference on Wearable and Implantable Body Sensor Networks (BSN), 2012, pp. 130–135.

[22] G. Douglas, C. Corcoran, S. Pavey, *Network 1000* Opinions and Circumstances of Visually Impaired People in Great Britain: Report based on Over 1000 Interviews, Visual Impairment Centre for Teaching and Research, University of Birmingham, 2006.

[23] J. Engel, T. Schöps, D. Cremers, LSD-SLAM: large-scale direct monocular SLAM, in: European Conference on Computer Vision (ECCV), 2014, pp. 834–849.

[24] N. Engelhard, F. Endres, J. Hess, J. Sturm, W. Burgard, Real-time 3D visual SLAM with a hand-held RGB-D camera, in: Proceedings of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden, vol. 180, 2011.

[25] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.

[26] D. Filliat, A visual bag of words method for interactive qualitative localization and mapping, in: IEEE International Conference on Robotics and Automation (ICRA), 2007, pp. 3921–3926.

[27] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, G. Wyeth, Openfabmap: an open source toolbox for appearance-based loop closure detection, in: The International Conference on Robotics and Automation, IEEE, St Paul, Minnesota, 2011.

[28] M. Hafez, Tactile interfaces: technologies, applications and challenges, Vis. Comput. 23 (4) (2007) 267–272.

[29] J. Hartcher-O'Brien, M. Auvray, V. Hayward, Perception of distance-to-obstacle through time-delayed tactile feedback, in: World Haptics Conference (WHC), IEEE, 2015, pp. 7–12.

[30] H. Hile, A.L. Liu, G. Borriello, R. Grzeszczuk, R. Vedantham, J. Kosecka, Visual navigation for mobile devices., IEEE MultiMedia 17 (2) (2010) 16–25.

[31] J. Hu, C. Chang, N. Tardella, J. Pratt, J. English, et al., Effectiveness of haptic feedback in open surgery simulation and training systems, in: J.W. Westwood, et al. (Eds.), Medicine Meets Virtual Reality, vol. 14, 2006, pp. 213–218.

[32] R. Huitl, F. Reinshagen, S. Hilsenbeck, G. Schroth, NavVis 3D Mapping Trolley, 2014.

[33] A.K. Jain, F. Farrokhnia, Unsupervised texture segmentation using Gabor filters, in: IEEE International Conference on Systems, Man and Cybernetics, 1990, 1990, pp. 14–19.

[34] A.A. Kalia, G.E. Legge, N.A. Giudice, Learning building layouts with non-geometric visual information: the effects of visual impairment and age, Perception 37 (11) (2008) 1677.

[35] G. Klein, D. Murray, Parallel tracking and mapping on a camera phone, in: Eighth IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, 2009, pp. 83–86.

[36] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.

[37] K. Konolige, M. Agrawal, Frame–frame matching for realtime consistent visual mapping, in: IEEE International Conference on Robotics and Automation, 2007, pp. 2803–2810.

[38] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.

[39] J.S. Lee, S. Lucyszyn, A micromachined refreshable Braille cell, J. Microelectromech. Syst. 14 (4) (2005) 673–682.

[40] J.J. Liu, C. Phillips, K. Daniilidis, Video-based localization without 3D mapping for the visually impaired, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2010, pp. 23–30.

[41] D.G. Lowe, Object recognition from local scale-invariant features, in: IEEE International Conference on Computer Vision (ICCV), vol. 2, 1999, pp. 1150–1157.

[42] S. Maidenbaum, S. Levy-Tzedek, D.-R. Chebat, A. Amedi, Increasing accessibility to the blind of virtual environments, using a virtual mobility aid based on the "eyecane", PloS One 8 (8) (2013) e72555.

[43] R. Manduchi, S. Kurniawan, Mobility-related accidents experienced by people with visual impairment, AER J.: Res. Pract. Vis. Impairment Blindness 4 (2) (2011) 44–54.

[44] M.J. Milford, G.F. Wyeth, SeqSLAM: visual route-based navigation for sunny summer days and stormy winter nights, in: IEEE International Conference on Robotics and Automation (ICRA), 2012, pp. 1643–1649.

[45] MIPsoft, Blindsquare.com, 2015.

[46] J. Neira, A.J. Davison, J.J. Leonard, Guest editorial special issue on visual SLAM, IEEE Trans. Robotics 24 (5) (2008) 929–931.

[47] Q.-H. Nguyen, H. Vu, Q.-H. Nguyen, et al., Mapping services in indoor environments based on image sequences, in: IEEE International Conference on Communications and Electronics (ICCE), 2014a, pp. 446–451.

[48] Q.-H. Nguyen, H. Vu, T.-H. Tran, D. Van Hamme, P. Veelaert, W. Philips, Q.-H. Nguyen, A visual SLAM system on mobile robot supporting localization services to visually impaired people, in: Proceedings of the European Conference on Computer Vision (ECCV) 2014 Workshops, Springer, 2014b, pp. 716–729.

[49] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 2161–2168.

[50] OpenSignal, Global State of LTE Report, February 2013, Technical Report, 2013.

[51] J. Pasquero, V. Hayward, Stress: a practical tactile display system with one millimeter spatial resolution and 700 Hz refresh rate, in: Proceedings Eurohaptics, vol.2003, 2003, pp. 94–110.

[52] T. Pey, F. Nzegwu, G. Dooley, Functionality and the Needs of Blind and Partially-sighted Adults in the UK: An Interim Report, Technical Report, Guide Dogs for the Blind Association, 2006.

[53] I. Poupyrev, S. Maruyama, J. Rekimoto, Ambient touch: designing tactile interfaces for handheld devices, in: Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology, 2002, pp. 51–60.

[54] L. Ran, S. Helal, S. Moore, Drishti: an integrated indoor/outdoor blind navigation system and service, in: IEEE Conference on Pervasive Computing and Communications, 2004. PerCom 2004, 2004, pp. 23–30.

[55] J. Rivera-Rubio, I. Alexiou, A.A. Bharath, RSM Dataset, 2014, http://rsm.bicv.org.

[56] J. Rivera-Rubio, I. Alexiou, A.A. Bharath, Appearance-based indoor localization: a comparison of patch descriptor performance, Pattern Recognit. Lett. (2015).

[57] J. Rivera-Rubio, I. Alexiou, L. Dickens, R. Secoli, E. Lupu, A.A. Bharath, Associating locations from wearable cameras, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2014, pp. 1–13.URL http://www.bmva.org/bmvc/2014/.

[58] RNIB, RNIB Strategy Focus and Goals 2009–2014, Technical Report, Royal National Institute of Blind People, 2009.

[59] RNIB, Living with Sight Loss Hackathon, 2012.

[60] U.R. Roentgen, G.J. Gelderblom, M. Soede, L.P. de Witte, Inventory of electronic mobility aids for persons with visual impairments: a literature review, J. Vis. Impairment Blindness 102 (11) (2008) 702–724.

[61] T. Schöps, J. Engel, D. Cremers, Semi-dense visual odometry for AR on a smartphone, in: InMixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on 2014, IEEE, 2014, pp. 145–150.

[62] G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, E. Steinbach, Mobile visual location recognition, IEEE Signal Process. Mag. 28 (4) (2011) 77–89.

[63] Sendero Group LLC, StreetTalk VIP GPS, 2015.
[64] S.J. Simske, Meta-algorithmics: Patterns for Robust, Low Cost, High Quality Systems, John Wiley & Sons, 2013.
[65] N. Snavely, S.M. Seitz, R. Szeliski, Photo tourism: exploring photo collections in 3D, in: ACM Transactions on Graphics (TOG), 2006, pp. 835–846.
[66] L. Spirkovska, Summary of Tactile User Interfaces Techniques and Systems, Technical Report, NASA Ames Research Center, Moffett Field, CA, United States, 2005.
[67] H. Strasdat, J. Montiel, A.J. Davison, Scale drift-aware large scale monocular SLAM., in: Robotics: Science and Systems, vol. 2, 2010, p. 5.
[68] B. Tsuji, G. Lindgaard, A. Parush, Landmarks for navigators who are visually impaired, in: Proceedings of XXII International Cartographic Conference A Coruña 2005 Proceedings, 2005.
[69] B. Tversky, Some ways that maps and diagrams communicate, in: C. Freksa, et al. (Eds.), Lecture Notes in Computer Science, vol. LNAI 1849, 2000, pp. 72–79.
[70] A. Vedaldi, B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008.
[71] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 34 (3) (2012) 480–492.
[72] J. Ventura, C. Arth, G. Reitmayr, D. Schmalstieg, Global localization from monocular SLAM on a mobile phone, IEEE Trans. Visual. Comput. Graph. 20 (4) (2014) 531–539.
[73] F. Vidal-Verdú, R. Navas-González, Thermopneumatic actuator for tactile displays, in: 18th Conference on Design of Circuits and Integrated Systems, DCIS, 2003, pp. 629–633.
[74] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, R.R. Choudhury, No need to war-drive: unsupervised indoor localization, in: International Conference on Mobile Systems, Applications, and Services, 2012, pp. 197–210.
[75] J.S. Warner, R.G. Johnston, GPS spoofing countermeasures, Homeland Secur. J. 25 (2) (2003) 19–27.
[76] T.P. Weldon, W.E. Higgins, D.F. Dunn, Efficient Gabor filter design for texture segmentation, Pattern Recognit. 29 (12) (1996) 2005–2015.
[77] J. Worsfold, E. Chandler, Wayfinding Project, Technical Report, Royal National Institute of Blind People, London, 2010.
[78] M. Yang, L. Zhang, S.C. Shiu, D. Zhang, Gabor feature based robust representation and classification for face recognition with Gabor occlusion dictionary, Pattern Recognit. 46 (7) (2013) 1865–1878.