Pattern Recognition Letters 66 (2015) 109–117

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Appearance-based indoor localization: A comparison of patch descriptor performance[☆]

Jose Rivera-Rubio[*], Ioannis Alexiou, Anil A. Bharath

Imperial College London, South Kensington Campus, London SW7 2AZ, United Kingdom

ARTICLE INFO

Article history:
Available online 18 March 2015

Keywords:
Visual localization
Descriptors
Human–computer interaction
Assistive devices

ABSTRACT

Vision is one of the most important of the senses, and humans use it extensively during navigation. We evaluated different types of image and video frame descriptors that could be used to determine distinctive visual landmarks for localizing a person based on what is seen by a camera that they carry. To do this, we created a database containing over 3 km of video-sequences with ground-truth in the form of distance travelled along different corridors. Using this database, the accuracy of localization—both in terms of knowing which route a user is on—and in terms of position along a certain route, can be evaluated. For each type of descriptor, we also tested different techniques to encode visual structure and to search between journeys to estimate a user's position. The techniques include single-frame descriptors, those using sequences of frames, and both color and achromatic descriptors. We found that single-frame indexing worked better within this particular dataset. This might be because the motion of the person holding the camera makes the video too dependent on individual steps and motions of one particular journey. Our results suggest that appearance-based information could be an additional source of navigational data indoors, augmenting that provided by, say, radio signal strength indicators (RSSIs). Such visual information could be collected by crowdsourcing low-resolution video feeds, allowing journeys made by different users to be associated with each other, and location to be inferred without requiring explicit mapping. This offers a complementary approach to methods based on simultaneous localization and mapping (SLAM) algorithms.

1. Introduction

Satellite-based global positioning systems (GPS) have been available to consumers for many years. When combined with other sensor data, such as terrestrial-based radio signal strength indicators (RSSI), the quality of pedestrian localization within cities, at street level, can be quite reliable. Recently, interest has been gathering for the development of systems for indoor position sensing: we might consider this to be the next challenge in localization systems [15,17,28,33]. Indoor sensing is likely to require additional infrastructure, such as Bluetooth-based RSSI, or time-of-flight systems. At the time of writing, both of these are reported to be under trial.

Despite this, vision-based navigation systems are under active development. This might be because such systems do not require special markers to be embedded within the environment. However, another reason could be that vision provides an immensely rich source of data, from which estimating position is also possible. For example, in emerging applications [22] such as mobile domestic robots, merely knowing where a robot is not enough: a robot often needs some information about its immediate environment in order to take appropriate decisions. This includes navigating around obstacles and identifying important objects (e.g. pets).

Systems and devices that are designed to help humans to navigate would be improved by incorporating vision as one of the sensing modalities. This is particularly true of systems that are designed to help visually impaired people to navigate (assistive navigation systems). However, for wearable or smartphone-based systems, accuracy and power consumption remain two of the challenges to the reliable and continuous use of computer vision techniques. Visual localization accuracy is affected by several factors, including the techniques used to infer location from visual data. In the case of feature-based SLAM [9], for example, a lack of features, or a highly occluded environment, can reduce accuracy.

Wang et al. [37] have recently suggested an interesting approach to localization based on the principle of identifying landmarks in ambient signals. These ambient signals are acquired from a crowdsourcing-like approach, rather than explicitly mapping out signal strength and

[☆] This paper has been recommended for acceptance by Gabriella Sanniti di Baja.

[*] Corresponding author. Tel.: +44 20 7594 5463.
E-mail address: jose.rivera@imperial.ac.uk (J. Rivera-Rubio).

http://dx.doi.org/10.1016/j.patrec.2015.03.003

WiFi identifiers and appears to offer good performance, with median absolute localization error of less than 1.7 m. Perhaps more importantly, it removes the need to change building infrastructure specifically for localization. One way to strengthen the landmark approach would be to incorporate *visual* cues, automatically mined from the video data. Theoretically speaking, such an approach might be limited by (i) the quality of the image acquisition, which could be affected by blur, poor focusing, inadequate resolution or poor lighting; (ii) the presence of occlusions to otherwise stable visual landmarks; (iii) visual ambiguity: the presence of visually-similar structures, particularly within man-made environments.

We now consider something of an ideal situation in which we can harvest visual signatures from several journeys down the same route; the approach starts with the idea of collecting *visual paths*, and using the data from these to localize the journeys of users relative to each other, and to start and end points.

2. Visual paths

Consider two users, Juan and Mary, navigating at different times along the same notional path. By notional path, we refer to a route that has the same start and end points. An example of indoor notional paths would be the navigation from one office to another, or from a building entrance to a reception point. For many buildings, such notional paths might allow different *physical* trajectories which could diverge. For example, users might take either stairs or lifts, creating path splits and merges. Such complex routes could be broken down into route (or path) segments, and path segments could contribute to more than one complete notional path.

For any notional path or path segment, both humans and autonomous robots would “experience” a series of cues that are distinctive when navigating along that path. In some instances, however, the cues might be ambiguous, just as they might be for radio signal strength indicators, audio cues and other environmental signals. A vision-based system would need to analyze the visual structure in sequences from hand-held or wearable cameras along path segments in order to answer two questions: which notional path or segment is being navigated, and where along a specific physical path, relative to start and end point, a person might be. We addressed the first of these questions in previous work [30].

Returning to the two-user scenario, let us assume that Juan has been the first to navigate along the path, and has collected a sequence of video frames during his successful navigation. As Mary makes her way along the path, we wish to be able to associate the images taken by Mary with those taken by Juan (see Fig. 1). The ability to do this allows us to locate Mary relative to the journey of Juan *from the visual data acquired by both*. For only two users, this may seem an uninteresting thing to do. However, imagine that this association is done between not two, but multiple users, and is applied to several physical paths that together form the navigable space of a building. Achieving this association would enable some types of inference to be performed. In particular:

- The visual path data would be a new source of data that could be used for location estimation;
- The association of image locations would allow visual change detection to be performed over many journeys along the same route, made at different times;
- Through non-simultaneous, many-camera acquisition, one could achieve more accurate mapping of a busy space, particularly where moving obstructions might be present;
- Visual object recognition techniques could be applied to recognize the nature of structures encountered along the route, such as exits, doorways and so on.

Using continuously acquired images provides a new way for humans to interact with each other through establishing associations between the visual experiences that they have shared, independent of any tags that have been applied. The concept is illustrated in Fig. 2(a). In this diagram, four users are within the same region of a building; however, two pairs of users (A,C) and (B,D) are associated with having taken similar trajectories to each other. With a sufficient number of users, one could achieve a crowdsourcing of visual navigation information from the collection of users, notional paths and trajectories.

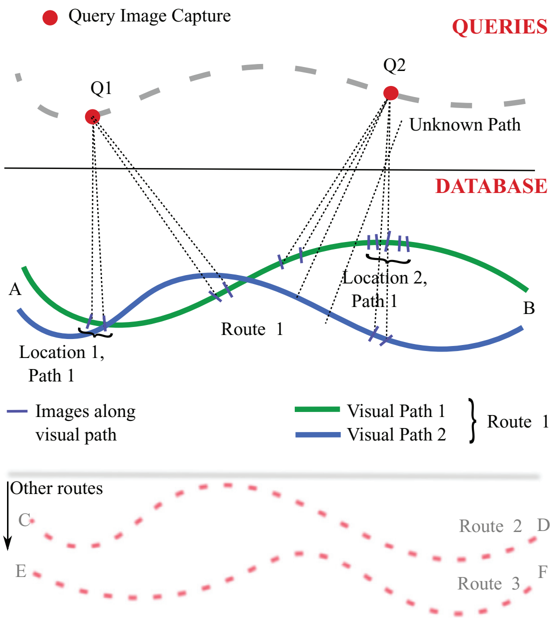


Fig. 1. The idea behind searching across data from navigators of the same physical path: after navigating the space twice, Juan’s visual path data (A, B) is indexed and stored in a database. Mary enters the same space (*unknown path*), and the images acquired as she moves are compared against the visual path of Juan, providing a journey-centric estimate of location. With many journeys collated, location can be inferred with respect to the pre-collected paths in the database.

One intriguing possibility would be to provide information to visually-impaired users. For example, in an assistive system, the visual cues that sighted individuals experience along an indoor journey could be mined, extracting reliable information about position and objects (e.g. exit signs) that are of interest. While other sources of indoor positioning information, such as the locations of radio beacons, can aid indoor navigation, some visual cues are likely to be stable over long periods of time, and do not require extra infrastructure beyond that already commonly installed. Collecting distinctive visual cues over many journeys allows stable cues to be learned. Finally, in contrast to signal-based methods of location landmarks [37], the “debugging” of this type of navigation data—i.e. images, or patches within images—is uniquely human-readable: it can be done simply through human observation of what might have visibly changed along the path. Perhaps most compelling of all, visual path data can be acquired merely by a sighted user sweeping the route with a hand-held or wearable camera.

3. Vision-based approaches to navigation

The current state-of-the-art methods for robot navigation make use of simple visual features and realistic robot motion models in order to map, then to navigate. For human navigation, the challenge is slightly greater, due partly to the variability of human motion. Nevertheless, recent progress in simultaneous localization and mapping (SLAM) [24] and parallel tracking and mapping (PTAM) [19] have yielded stunning results in producing geometric models of a physical,

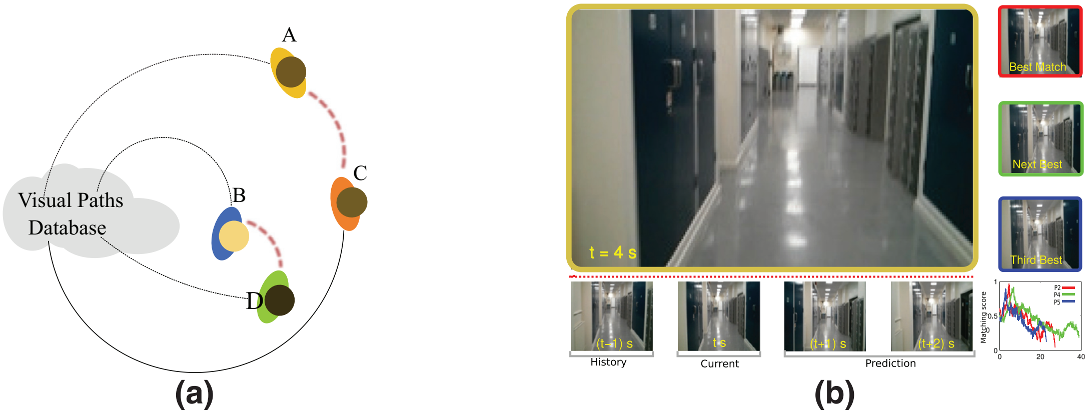


Fig. 2. In (a), we illustrate the concept of using the visual path database to establish rough correspondence in users' locations through their visual experiences. Users (A,C) and (B,D) experience similar visual paths (see the text for details). In (b), the current view captured by a camera and views from the best match *paths* that have been captured through that space, to the immediate right. The first four bottom panels show current, historical, and predicted images, based on the query from the best matching visual path. The right, bottom image shows the similarity scores from other journeys taken along the same notional path. The techniques that enable this type of match to be done are discussed in Section 5.

recovering geometry and camera pose simultaneously from hand-held devices.

At the same time, being able to recognize certain objects while performing SLAM could improve accuracy, reducing the need for loop closure and allowing better—more reliable—self-calibration [32]. Recognition pipelines in computer vision have recently taken great strides, both in terms of scalability and accuracy. Thus, the idea of collaboratively mapping out a space through wearable or hand-held cameras is very attractive.

Appearance-based navigation, closely related to navigation, has been reported as one of many mechanisms used in biology, and has been explored by various groups in different animals (see, for example, [6,10,13]). Appearance-based approaches can add to the information gained using SLAM-type algorithms. Indeed, in a robust system, we might expect several sources of localization information to be employed. Consider, for example, outdoor navigation in cities: GPS can be combined with WiFi RSSI. Doing so improves overall accuracy, because the errors in these two localization systems are unlikely to be highly correlated over relatively short length scales (≈100 m), and would only be trivially correlated (but highly) over longer distances. Localization systems often rely on motion models embedded into tracking algorithms, such as Kalman, extended Kalman [9] filtering, or particle-filtering [27], to infer position. More recently, general purpose graphics processing units (GP-GPUs) have enabled camera position to be quickly and accurately inferred relative to a point cloud by registering whole images with dense textured models [24].

Anecdotal evidence and conversations with UK groups supporting visually-impaired people suggests that no *single* source of data or single type of algorithm will be sufficient to meet the needs of users who are in an unfamiliar space, or who might suffer from visual impairment. It is likely that a combination of sensors and algorithms is called for.

3.1. A biological perspective

Research into the mechanisms employed by humans during pedestrian navigation suggests that multisensory integration plays a key role [25]. Indeed, studies into human spatial memory using virtual reality and functional neuroimaging [3,4] suggest that the human brain uses a combination of representations to self-localize that might be termed as *allocentric* and *egocentric*. The egocentric representation supports identifying a location based on sensory patterns recognized from previous experiences in a given location. Allocentric representations use a reference frame that is independent of one's location. The respective coordinate systems can, of course, be interchanged via simple transformations, but the sensory and cognitive processes underlying navigation in both cases are thought to be different.

The two forms of representation are typified by different types of cells, and, in some cases, different neuronal signal pathways. Within some mammals, such as mice, it appears that a multitude of further sub-divisions of computational mechanisms lie behind location and direction encoding. For example, in the hippocampus, there are at least four classes [14] of encoding associated with position and heading. Hippocampal *place cells* display elevated firing when the animal is in a particular location [11]. The environmental cues that affect hippocampal place cells include vision and odour, so the inputs to these cells are not necessarily limited to any one type of sensory input.

Grid cells, on the other hand, show increased firing rates when the animal is present at a number of locations on a spatial grid; this suggests that some form of joint neuronal encoding is at work, and, indeed, there is some evidence that place cell responses arise through a combination of grid cells of different spacing [23]. Boundary cells in the hippocampus appear to encode just that: the distance to the boundaries of a spatial region. This encoding seems to be relative to the direction the animal is facing but independent of the relation between the animal's head and body; they are therefore, examples of an allocentric scheme.

In conclusion, biology seems to employ not only several sensory inputs to enable an organism to locate itself relative to the environment, but also different computational mechanisms. The evidence of these multiple strategies for localization and navigation [14,26] motivates the idea for an appearance-based localization algorithm.

4. The dataset

A total of 60 videos were acquired from six corridors of the RSM building at Imperial College London. Two different devices were used. One was a LG Google Nexus 4 mobile phone running Android 4.4.2 "KitKat". The video data were acquired at approximately 24–30 fps at two different acquisition resolutions, corresponding to 1280 × 720 and 1920 × 1080 pixels. The other device was a wearable Google Glass (2013 Explorer edition) acquiring data at a resolution of 1280 × 720, and a frame rate of around 30 fps. A surveyor's wheel (Silverline) with a precision of 10 cm and error of ±5% was used to record distance, but was modified by connecting the encoder to the general purpose input/output (GPIO) pins of a Raspberry Pi running a number of measurement processes. The Pi was synchronized to network time using network time protocol (NTP), enabling synchronization with timestamps in the video sequence. Because of the variable frame rate of acquisition, timestamp data from the video were used to align

Table 1
Detailed summary of the RSM dataset: L length of the corridors and Fr number of video frames. The single frames are representative images from hand-held videos of selected corridors of the RSM building at Imperial College London. The dataset includes both hand-held and wearable camera examples, all containing ground-truth location relative to distance traversed along labelled paths. The grand totals are: $L = 3042$ km and #Fr = 90,302 frames.

$\bar{L}$(m)	57.9	31.0	52.7	49.3	54.3	55.9
$\sum L$	585.6	312.4	524.2	497.7	562.0	560.4
$\bar{\text{Fr}}$	2157	909	1427	1583	1782	1471
$\sum$ Fr	19,379	9309	14,638	15,189	16,823	14,964

ground-truth measurements with frames. This data are used to assess the accuracy of estimating positions, and not for any form of training.

In total, 3.05 km of data are contained in this dataset, at a natural indoor walking speed. For each corridor, 10 passes (i.e. 10 separate visual paths) are obtained; five of these are acquired with the hand-held Nexus, and the other five with Glass. Table 1 summarizes the acquisition. As can be seen, the length of the sequences varies within some corridors; this is due to a combination of different walking speeds and/or different frame rates. Lighting also varied, due to a combination of daylight/nighttime acquisitions, and occasional windows acting as strong lighting sources in certain sections of the building. Changes were also observable in some videos from one pass to another due to the presence of changes (path obstructions being introduced during a cleaning activity) and occasional appearances of people.

In total, more than 90,000 frames of video were labelled with positional ground-truth. The dataset is publicly available for download at http://rsm.bicv.org [29].

5. Methods: Indexing

We evaluated the performance of different approaches to query images taken from one visual path against others stored in the database. In order to index and query visual path datasets, we used the steps illustrated in Fig. 3. The details behind each of the steps (e.g. gradient estimation, spatial pooling) are described in the remainder of this section. They include techniques that operate on *single* frames as well as descriptors that operate on *multiple* frames, at the frame level and at the patch level. All the performance evaluation experiments were carried out at low-resolution (208×117 pixels) versions of the sequences, keeping bandwidth and processing requirements small.

5.1. Frame-level descriptor

Based on the use of optical flow in motion estimation [39] and space-time descriptors in action recognition [38] we estimated in-plane motion vectors using a simple approach. We first applied derivative filters along (x, y, t) dimensions, yielding a 2D+t, i.e. spatio-temporal, gradient field. To capture variations in chromatic content from the visual sequence, we computed these spatio-temporal gradients separately for each of the three RGB channels of the preprocessed video sequences. This yielded a 3×3 matrix at each point in space. Temporal smoothing was applied along the time dimension, with a support of 11 neighbouring frames. Finally, the components of the matrix were each averaged (pooled) over 16 distinct spatial regions, not very dissimilar to those to be described later in this paper. For each visual path, this yielded 144 signals, of length approximately equal to the video sequences. An illustration of the time series for one visual path is shown in Fig. 4.

At each point in time, the values over the 144 signal channels are also captured into a *single* space-time descriptor per frame: LW_COLOR. Our observations from the components of this descriptor are that (a) relative ego-motion is clearly identifiable in the signals; (b) stable patterns of motion may also be identified, though changes in the precise trajectory of a user could also lead to perturbations in these signals, and hence to changes in the descriptor vectors. Minor changes in trajectory might, therefore, reduce one's ability to match descriptors between users. These observations, together with the possibility of partial occlusion, led us to the use of *patch* based descriptors, so that multiple descriptors would be produced for each frame. These are introduced next.

5.2. Patch-level descriptors

The patch descriptors can be further divided into two categories: those produced from patches of single frames, and those that are based on patches acquired over multiple frames; the latter are *space-time* patch descriptors. We explored two distinct single-frame descriptors, and three distinct space-time descriptors. We first describe the single-frame descriptors.

5.2.1. Spatial patch descriptors (single-frame)

The spatial patch descriptors consist of the Dense-SIFT descriptor [20,21,34] and a tuned, odd-symmetric Gabor-based descriptor. The SIFT descriptor was calculated by dense sampling of the smoothed estimate of $\vec{\nabla} f(x, y; \sigma)$ where $f(x, y; \sigma)$ represents the scale-space embedding of image $f(x, y)$ within a Gaussian scale-space at scale σ.

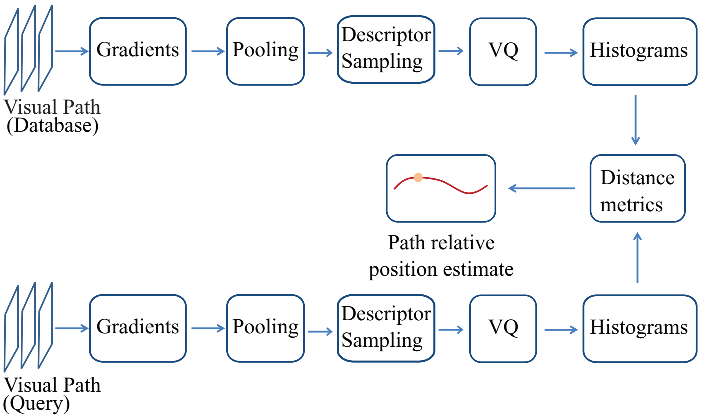


Fig. 3. This diagram illustrates the stages in processing the sequences in the database and for the queries. This diagram does not show the process behind the estimation of ground-truth for the experiments, which is described separately in Section 4. Variants of the gradient operators, pooling operators, quantization and distance metrics are described in Section 5.

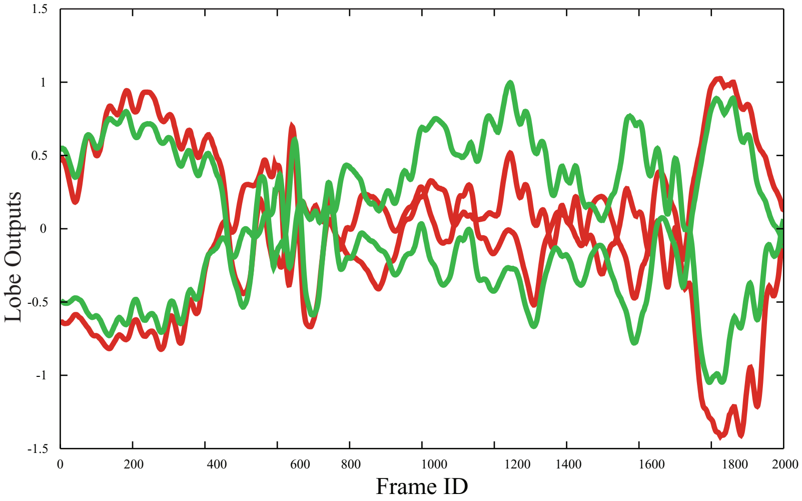


Fig. 4. Four (of 144) representative signals acquired from a visual path; these signals encode changes in red and green channels as a user moves through space. The collection of signal traces at one point in time can be used to build a simple frame-level space-time descriptor: LW_COLOR. The signal amplitudes are spatially pooled temporal and spatial gradient intensities. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We used a standard implementation of dense SIFT from VLFEAT [34] with scale parameter, $\sigma \approx 1$, and with a stride length of 3 pixels. This yielded around 2000 descriptors per frame, each describing a patch of roughly 10×10 pixels in the frame.

We compared these with another single-frame technique devised in our lab: we used filters that we previously tuned on PASCAL VOC data [12] for image categorization. These consisted of 8-directional, 9×9 pixels spatial Gabor filters ($k = 1, \ldots, 8$; $\sigma = 2$). Each filter gives rise to a filtered image plane, denoted $\mathbf{G}_{k,\sigma}$. For each plane, we applied spatial convolution ($*$) with a series of pooling functions:

$$\mathbf{G}_{k,\sigma} * \Phi_{m,n} \quad (1)$$

where $\Phi_{m,n}$ is computed by spatial sampling of the pooling function:

$$\Phi(x, y; m, n) = \exp\left\{-\alpha\left[\log_e\left(\frac{x^2+y^2}{d_n^2}\right)\right]^2 - \beta|\theta - \theta_m|\right\} \quad (2)$$

with $\alpha = 4$ and $\beta = 0.4$. The values of $m = 0, \ldots, 7$ and $n = 0, 1, 2$ were taken to construct 8 regions at angles $\theta_m = m\frac{\pi}{4}$ for each of two distances $d_1 = 0.45$, $d_2 = 0.6$ away from the center of a spatial pooling region in the image plane. For the central region, corresponding to $m = 0$, there was no angular variation, but a log-radial exponential decay. This yielded a total of 17 spatial pooling regions. The resulting 17×8 fields were sub-sampled to produce a dense 136-dimensional descriptors, each representing an approximately 10×10 pixels image region. This resulted in, again, approximately 2000 descriptors per image frame after the result of Eq. (1) is sub-sampled. This is illustrated in Fig. 5.

5.2.2. Space-time patch descriptors

Given the potential richness available in the capture of space-time information, we explored three distinct approaches to generate space-time patch descriptors. These approaches all lead to multiple descriptors per frame, and all take into account neighbouring frames in time when generating the descriptor associated with each patch. Additionally, all three densely sample the video sequence. The three methods are (i) HOG 3D, introduced by Kläser et al. [18]; (ii) our space-time, antisymmetric Gabor filtering process (ST_GABOR); and (iii) our spatial derivative, temporal Gaussian (ST_GAUSS) filter.

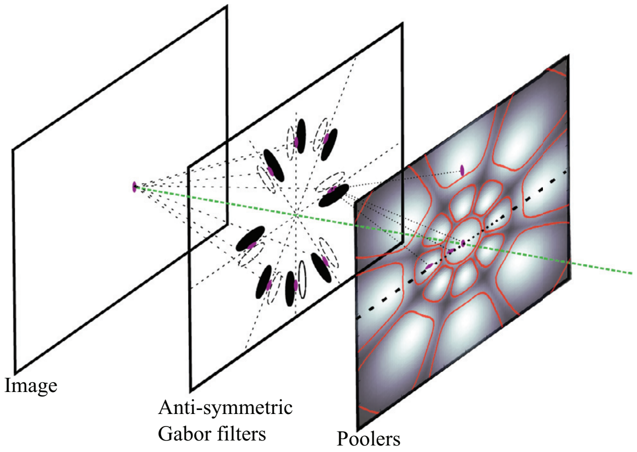


Fig. 5. Orientation selective masks are applied to the video frames by spatial convolution. The outputs of the masks are collected over space using the pooling functions. The outputs of the poolers are sub-sampled over the image plane to produce descriptors for indexing. See the text for further details.

(i) The HOG 3D descriptor (HOG3D) [18] was introduced to extend the very successful two-dimensional histogram of oriented gradients technique [8] to space-time fields, in the form of video sequences. HOG 3D seeks computational efficiencies by smoothing using box filters, rather than Gaussian spatial or space-time kernels. This allows three-dimensional gradient estimation across multiple scales using the *integral video* representation, a direct extension of the integral image idea [36]. The gradients from this operation are usually performed across multiple scales. We used the dense HOG 3D option from the implementation of the authors [18], and the settings yielded approximately 2000 descriptors per frame of video.

(ii) Space-time Gabor (ST_GABOR) functions have been used in activity recognition, structure from motion and other applications [2]. We performed one dimensional convolution between the video sequence and three one-dimensional Gabor functions along either one spatial dimension i.e. x or y, or along t. The one-dimensional convolution is crude, but appropriate if the videos have been downsampled. The spatial extent of the Gabor was set to provide one complete cycle of oscillation over approximately 5 pixels of spatial span, both for the x and y spatial dimensions. The filter for the temporal dimension was set to provide temporal support and one oscillation over approximately 9 frames. We also explored symmetric Gabor

Table 2
A summary of the different encoding methods and their relationships to different descriptors. The number of elements of each descriptor is also reported (**Dim**).

Method	ST	Dense	Dim	Encoding	Metric
SIFT	No	Yes	128	HA-4000 VLAD-256	χ^2 Hellinger
SF-GABOR	No	Yes	136	HA-4000 VLAD-256	χ^2 Hellinger
LW-COLOR	Yes	No	144	N/A	
ST_GABOR	Yes	Yes	221	HA-4000 VLAD-256	χ^2 Hellinger
ST_GAUSS	Yes	Yes	136	HA-4000 VLAD-256	χ^2 Hellinger
HOG3D	Yes	Yes	192	HA-4000 VLAD-256	χ^2 Hellinger

functions, but found them less favorable in early performance tests.

After performing three separate filtering operations, each pixel of each frame was assigned a triplet of values corresponding to the result of the each filtering operation. The three values were treated as being components of a 3D vector. Over a spatial extent of around 16 × 16 pixels taken at the central frame of the 9-frame support region, these vectors contribute weighted votes into 13 histogram bins according to their azimuth and elevations, with the weighting being given by the length of the vector. The votes were also partitioned into 17 regions according to the approximate spatial lobe pattern illustrated in Fig. 5, yielding a 221-dimension descriptor.

(iii) A final variant of space-time patch descriptor was designed. This consisted of spatial derivatives in space, combined with smoothing over time (ST_GAUSS). In contrast to the strictly one-dimensional filtering operation used for the space-time Gabor descriptor, we used two 5 × 5 gradient masks for the x and y directions based on derivatives of Gaussian functions, and an 11-point Gaussian smoothing filter in the temporal direction with a standard deviation of 2. Eight-directional quantization was applied to the angles of the gradient field, and a weighted gradient magnitude voting process was used to distribute votes across the 8 bins of a 136-dimensional descriptor. Like the ST_GABOR descriptor, pooling regions were created, similar to those shown in Fig. 5.

5.3. Frame-level encoding

Our initial conjecture was that whole frames from a sequence could be indexed compactly, using the single-frame descriptor (LW_COLOR). This was found to lead to disappointing performance (see Section 6). For the case of many descriptors-per-frame i.e. descriptors that are patch-based, we have the problem of generating around 2000 descriptors per frame, if dense sampling is used. Thus, we applied vector quantization (VQ) to the descriptors, then used histograms of VQ descriptors, effectively representing each frame as a histogram of words [7]. The dictionary was always built by excluding the entire journey from which queries are to be taken.

Two different approaches to the VQ of descriptors were taken, one based on standard k-means, using a Euclidean distance measure (hard assignment, "HA"), and one corresponding to the Vector of Locally Aggregated Descriptors (VLAD) [16]. For VLAD, a k-means clustering was first performed. For each descriptor, sums of residual vectors were used to improve the encoding. Further advances to the basic VLAD, which include different normalizations and multiscale approaches, are given by [1]. To compare encodings, either χ^2 or Hellinger distance metrics [35] were used to retrieve results for HA and VLAD encoding approaches respectively. Distance comparisons were performed directly between either hard assigned Bag-of-Words (BoW) or VLAD image encodings arising from collections of descriptors for each frame.

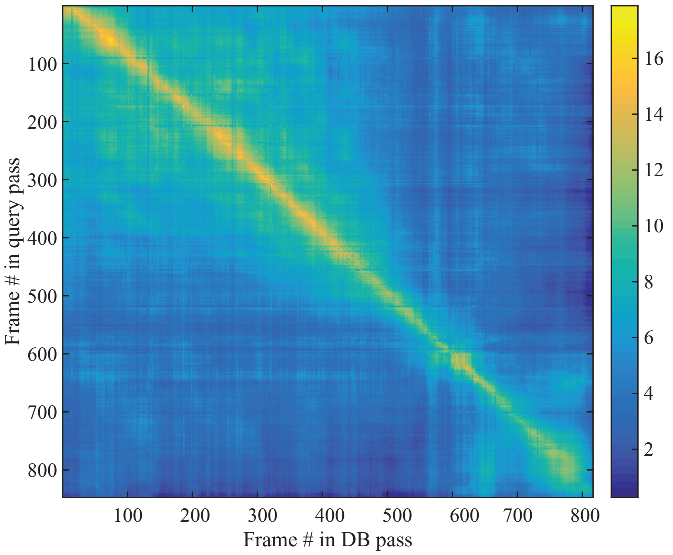


Fig. 6. Example of a χ^2 kernel produced by hard assignment and using the SF_GABOR descriptors when querying with pass 1 of corridor 2 against a database comprised of passes 2–10.

6. Experiments and results: Performance evaluation

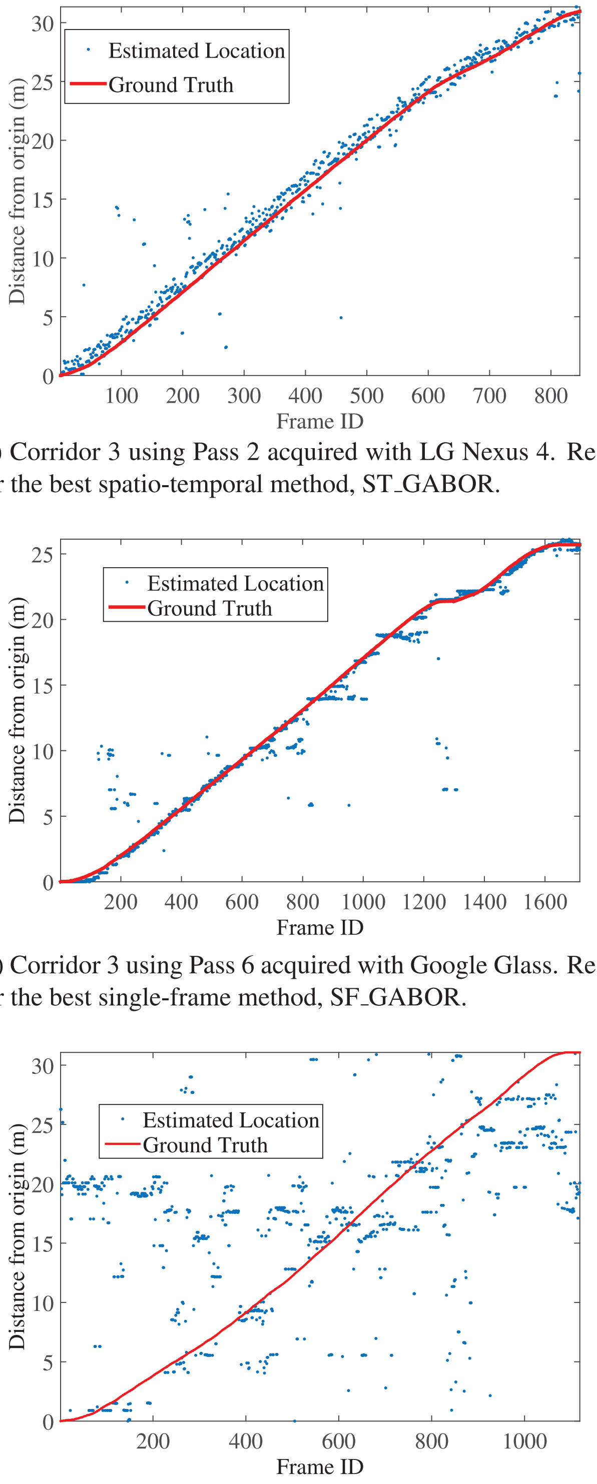
The methods for (a) describing spatial or space-time structure, (b) indexing and comparing the data are summarized in Table 2. The choice of parameters was selected to allow (a) as consistent a combination of methods as possible, allowing fair comparisons of the effect of one type of encoding or spatio-temporal operator to be isolated from others (b) to select parameter choices close to other research in the area, e.g. for image categorization, dictionary sizes of ≈256 and ≈4000 words are common.

6.1. Error distributions

Error distributions allow us to quantify the accuracy of being able to estimate *locations* along physical paths within the RSM dataset described in Section 4. To generate the error distributions, we did the following:

We started by using the kernels calculated in Section 5.3. One kernel is shown in Fig. 6, where the rows represent each frame from the query pass, and the columns represent each frame from one of the remaining database passes of that corridor. The values of the kernel along a row represent a "score" between a query and different database frames. In this experiment, we associated the position of the best match to the query frame, and calculated the error between this and the ground truth ϵ, in cm. In order to characterize the reliability of such scores, we performed bootstrap estimates of error distributions using 1 million trials. The distribution of the errors gives us a probability density estimate, from which we can get the cumulative distribution function (CDF) $P(x \le |\epsilon|)$. The outcome is shown in Fig. 8, where only the average across all the randomized samples is shown.

By permuting the paths that are held in the database and randomly selecting queries from the remaining path, we were able to assess the error distributions in localization. Repeated runs with random selections of groups of frames allowed the variability in these estimates to be obtained, including that due to different numbers of paths and passes being within the database.



(a) Corridor 3 using Pass 2 acquired with LG Nexus 4. Results for the best spatio-temporal method, ST_GABOR.

(b) Corridor 3 using Pass 6 acquired with Google Glass. Results for the best single-frame method, SF_GABOR.

(c) Corridor 2 using Pass 6 acquired with Google Glass. Results for the worst overall method, LW_COLOR.

Fig. 7. Estimated location vs. ground truth. Illustrative examples of good/bad location estimation performance (a) uses the best descriptor and a single-device dataset, (b) uses the best descriptor and a cross-device dataset and (c) uses the worst descriptor, and a multiple-device dataset.

If we consider the idea of crowdsourcing journey information from many pedestrian journeys through the same corridors, this approach to evaluating the error makes sense: all previous journeys could be indexed and held in the database; new journey footage would be submitted as a series of query frames (see Fig. 1).

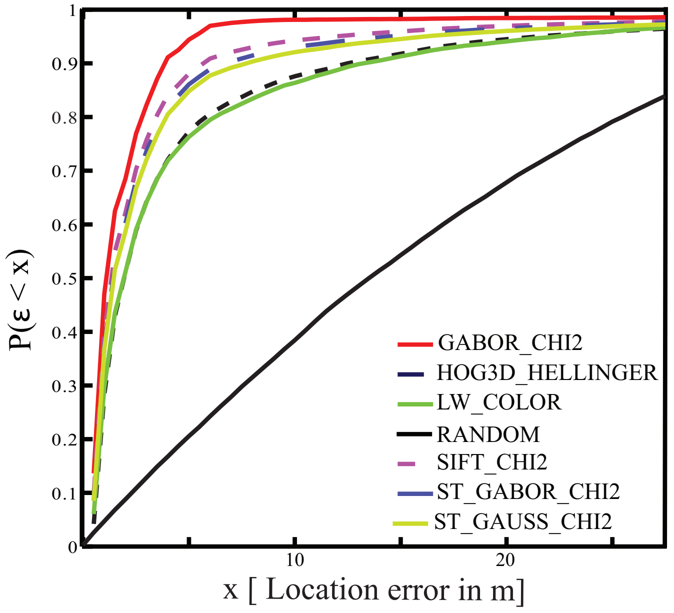


Fig. 8. Comparison of best performing version of all methods. The average CDFs were computed from the distributions generated after the random sampling described in Section 6.1. The results for a random test (RANDOM) were introduced as a "sanity check".

6.2. *Localization error vs ground-truth route positions*

As described in the previous section, by permuting the database paths and selecting, randomly, queries from the remaining path that was left out in the dictionary creation, we can assess the errors in localization along each corridor for each pass, and calculate, also, the average error in localization on a per-corridor basis, or per-path basis. For these, we used the ground-truth information acquired as described in Section 4. Fig. 7 provides some examples of the nature of the errors, showing evidence of those locations that are often confused with each other. As can be seen, for the better method (top trace of Fig. 7) while average errors might be small, there are, occasionally, large errors due to poor matching (middle trace). Errors are significantly worse for queries between different devices (see Fig. 7(c)).

Note that we did not use any tracking algorithms, and so there is no motion model or estimate of current location given the previous one. Incorporating a particle filter or Kalman filter should reduce the errors, particularly where there are large jumps within small intervals of time. This deliberate choice allows us to evaluate the performance of different descriptor and metric choices independently.

6.3. *Performance summaries*

We calculated the average of the absolute positional error (in cm) and the standard deviation of the absolute positional error in a subset of the complete RSM dataset (Table 3). We used a leave-one-journey-out approach (all the frames from an entire journey are excluded from the database). Using bootstrap sampling, we also estimated the cumulative density functions of the error distributions in position, which are plotted in Fig. 8. The variability in these curves is not shown, but is summarized in the last two columns of Table 3 through the area-under-curve (AUC) values. In the best case (SF_GABOR), AUCs of the order of 96% would mean errors generally below 2 m; in the worst (HOG3D), AUCs ≈ 90% would mean errors of around 5 m. These mean absolute error estimates are obtained as we permute the queries, the dictionary and the paths in the database.

Finally, we applied one implementation of the SLAM to this dataset, at the same frame resolution as for the appearance-based

Table 3
Summaries of average absolute positional errors and standard deviation of positional errors for different descriptor types and for different encoding methods (labelled by the corresponding metric used: χ^2 for HA and Hellinger for VLAD). μ_ϵ is the average absolute error, and σ_ϵ is the standard deviation of the error in cm. Single device case and in bold: best and worst AUC.

Method	Metric	Error summary (cm)		AUC (%)	
		μ_ϵ	σ_ϵ	Min	Max
SF_GABOR	χ^2	130.6	38.8	96.40	**96.75**
SF_GABOR	Hellinger	135.1	46.5	96.29	96.71
ST_GAUSS	χ^2	135.4	44.1	93.61	94.30
ST_GAUSS	Hellinger	144.1	52.4	92.69	93.47
ST_GABOR	χ^2	235.9	86.3	93.97	94.66
ST_GABOR	Hellinger	179.5	62.3	93.98	94.60
SIFT	χ^2	137.5	46.3	94.57	95.14
SIFT	Hellinger	132.7	41.4	94.34	94.95
HOG3D	χ^2	419.6	133.3	**90.89**	91.83
HOG3D	Hellinger	366.5	120.3	91.49	92.37
LW_COLOR	N/A	363.9	113.2	91.42	92.25

localization discussed in this paper. We chose the "EKF Mono SLAM" [5], which uses an extended Kalman filter (EKF) with 1-point RANSAC. We chose this implementation for three reasons: (a) it is a monocular SLAM technique, so comparison with the single-camera approach is fairer; (b) the authors of this package report error estimates—in the form of error distributions; and (c) the errors from video with similar resolutions (240 × 320) to ours were reported as being below 2 m for some sequences [5] in their dataset.

The results of the comparison were surprising, and somewhat unsatisfactory. The challenging ambiguity of the sequences in the RSM dataset, and possibly the low resolution of the queries, might explain the results. The feature detector, a FAST corner detector [31], produced a small number of features in its original configuration. We lowered the feature detection threshold until the system worked on a small number of frames from each sequence. Even with more permissive thresholds, the average number of FAST features averaged only 20 across our experiments. This small number of features led to inaccuracy in the position estimates, causing many of the experimental runs to stop when no features could be matched. The small number of features per frame is also not comparable with the feature density of the methods described in this paper, where an average of 2000 features per frame was obtained for the "dense" approaches. Dense SLAM algorithms might fare better.

7. Discussion

The performance comparisons shown in the cumulative error distributions of Fig. 8 would seem a fairly natural means of capturing localization performance. Yet, they do not suggest large differences in terms of the AUC metric (Table 3), given the large diversity in the complexity of the indexing methods. However, absolute position estimation errors tell a different story: average absolute errors are as high as 4 m for the worst performing method (HOG3D), and just over 1.3 m for the best performing method (SF_GABOR), if the same camera is used. The best performance compares very favorably with reported errors in positioning from multi-point WiFi signal strength measurements using landmark-based recognition that employs multiple (non-visual) sensing [33]. Indeed, it is very likely that the size of the errors we have observed can be reduced by incorporating simple motion models and a tracker, in the form of a Kalman filter.

A surprising result was that good levels of accuracy were obtained for images as small as 208 × 117 pixels. This suggests that relatively low-resolution cameras can be used to improve the performance of indoor localization systems. Being able to use such low resolutions of image reduces the indexing time, storage, power and bandwidth requirements.

8. Conclusion and future work

The advent of wearable and hand-held cameras makes appearance-based localization feasible. Interaction between users and their wearable device would allow for new applications such as localization, navigation and semantic descriptions of the environment. Additionally, the ability to crowdsource "visual paths" against which users could match their current views is a realistic scenario given ever improving connectivity.

We evaluated several types of descriptor in this retrieval-based localization scenario, achieving errors as small as 1.30 m over a 50 m distance of travel. This is surprising, given that we used low-resolution versions of our images, and particularly since our RSM dataset also contains very ambiguous indoor scenes.

We are currently working on enlarging the RSM database, by including larger numbers of journeys. A future goal will be to mitigate the effects of partial occlusion between different views of the same physical location. For example, face-detection might be applied to identify when and where people are within the scene acquired along a users' journey; we would avoid generating descriptors that covered these regions of image space. Other movable objects (chairs, trolleys) could also be actively detected and removed from indexing or queries.

The challenges associated with searching across video from multiple devices would still need to be solved. We can see from Section 6 that between-device queries have much higher error than within-device queries. This problem can be solved by either capturing and indexing data from a variety of devices for the same journeys, or by learning a mapping between devices. Another obvious strand of work would be to incorporate information from other sources, such as RSSI indicators, to reduce localization error.

Finally, we are exploring ways to combine the appearance-based technique described in this paper with SLAM and its variants. Doing this would allow geometric models from independent point-cloud sets to be associated with each other, allowing the continuous updating of the models that describe a physical space. Multiple geometric models, acquired from otherwise independent journeys, would support more detailed and reliable descriptions of an indoor, navigable space. It would also allow better interaction between the users of a building with its features, and with each other.

Our long-term goal is to convey the information acquired by sighted users to help people with visual impairment; this would require creating and updating rich descriptions of the visual and geometric structure of a physical space. This could be used in the making of indoor navigational aides, which would be rendered through haptic or audio interfaces, making the planning of journeys easier for the visually impaired.

Acknowledgements

Jose Rivera-Rubio's research is supported by an EPSRC DTP scholarship. The dataset hosting is supported by the EPSRC V&L Net Pump-Priming Grant 2013-1. We are also grateful to members of the Imperial College's Biologically Inspired Computer Vision group for their help and collaboration, and to Dr Riccardo Secoli and Dr Luke Dickens for their valuable input and useful discussions.

References

[1] R. Arandjelovic, A. Zisserman, All about vlad, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 1578–1585.
[2] M. Bregonzio, S. Gong, T. Xiang, Recognising action as clouds of space-time interest points, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009 (CVPR'09), IEEE, 2009, pp. 1948–1955.
[3] N. Burgess, Spatial memory: how egocentric and allocentric combine, Trend Cogn. Sci. 10 (2006) 551–557.

[4] N. Burgess, E.A. Maguire, J. O'Keefe, The human hippocampus and spatial and episodic memory, Neuron 35 (2002) 625–641.

[5] J. Civera, O.G. Grasa, A.J. Davison, J.M.M. Montiel, 1pointt ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry, J. Field Robot. 27 (2010) 609–631.

[6] M. Collett, How desert ants use a visual landmark for guidance along a habitual route, Proc. Natl. Acad. Sci. 107 (2010) 11638–11643.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision, ECCV, Prague, 2004, pp. 1–2.

[8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005 (CVPR'05), IEEE, 2005, pp. 886–893.

[9] A.J. Davison, Real-time simultaneous localisation and mapping with a single camera, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2003, pp. 1403–1410.

[10] D.A. Dombeck, C.D. Harvey, L. Tian, L.L. Looger, D.W. Tank, Functional imaging of hippocampal place cells at cellular resolution during virtual navigation, Nat. Neurosci. 13 (2010) 1433–1440.

[11] G. Dragoi, S. Tonegawa, Selection of preconfigured cell assemblies for representation of novel spatial experiences, Phil. Trans. Roy. Soc. B: Biol. Sci. 369 (2014) 20120522.

[12] M. Everingham, L. Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes (VOC) challenge, Int. J. Comput. Vis. 88 (2009) 303–338.

[13] S. Fry, R. Wehner, Look and turn: Landmark-based goal navigation in honey bees, J. Exp. Biol. 208 (2005) 3945–3955.

[14] T. Hartley, C. Lever, N. Burgess, J. O'Keefe, Space in the brain: How the hippocampal formation supports spatial cognition, Phil. Trans. Roy. Soc. B: Biol.l Sci. 369 (2014) 20120510.

[15] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, E. Steinbach, TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping, in: 19th IEEE International Conference on Image Processing (ICIP), IEEE, 2012, pp. 1773–1776.

[16] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 1704–1716.

[17] W. Kadous, S. Peterson, Indoor maps: The next frontier, in: Google IO, 2013.

[18] A. Kläser, M. Marszalek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: British Machine Vision Conference (BMVC), British Machine Vision Association (BMVA), 2008, pp. 995–1004.

[19] G. Klein, D. Murray, Parallel tracking and mapping on a camera phone, in: 8th IEEE International Symposium on Mixed and Augmented Reality, 2009 (ISMAR'09), IEEE, 2009, pp. 83–86.

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

[21] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, IEEE, 1999, pp. 1150–1157.

[22] A. Möller, M. Kranz, R. Huitl, S. Diewald, L. Roalter, A mobile indoor navigation system interface adapted to vision-based localization, in: Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia, ACM, 2012, p. 4.

[23] E.I. Moser, E. Kropff, M.B. Moser, Place cells, grid cells, and the brain's spatial representation system, Annu. Rev. Neurosci. 31 (2008) 69–89.

[24] R.A. Newcombe, S.J. Lovegrove, A.J. Davison, Dtam: Dense tracking and mapping in real-time, in: 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 2320–2327.

[25] P. Panagiotaki, S. Lambrey, A. Berthoz, Neural bases and cognitive mechanisms of human spatial memory, Image. Spat. Cogn. 66 (2006) 85–100.

[26] B. Poucet, F. Sargolini, E.Y. Song, B. Hangya, S. Fox, R.U. Muller, Independence of landmark and self-motion-guided navigation: a different role for grid cells, Phil. Trans. Roy. Soc. B: Biol. Sci. 369 (2014) 20130370.

[27] M.L. Pupilli, a.D. Calway, Real-time camera tracking using a particle filter, in: Proceedings of the British Machine Vision Conference, British Machine Vision Association (BMVA), 2005, pp. 50.1–50.10.

[28] M. Quigley, D. Stavens, Sub-meter indoor localization in unmodified environments with inexpensive sensors, in: Intelligent Robots and Systems, IEEE, 2010, pp. 2039–2046.

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

[30] J. Rivera-Rubio, S. Idrees, I. Alexiou, L. Hadjilucas, A.A. Bharath, Mobile visual assistive apps: Benchmarks of vision algorithm performance, in: A. Petrosino, L. Maddalena, P. Pala (Eds.), Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2013, pp. 30–40.

[31] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: European Conference on Computer Vision, Springer, 2006, pp. 430–443.

[32] R.F. Salas-Moreno, R.a. Newcombe, H. Strasdat, P.H. Kelly, A.J. Davison, SLAM++: Simultaneous localisation and mapping at the level of objects, 2013 IEEE Conference on Computer Vision and Pattern Recognition (2013) 1352–1359.

[33] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, Y. Zhang, Walkie-markie: indoor pathway mapping made easy, in: Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, USENIX Association, 2013, pp. 85–98.

[34] A. Vedaldi, B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008.

[35] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 480–492.

[36] P. Viola, M. Jones, Robust real-time object detection, Int. J. Comput. Vis. 4 (2001) 34–47.

[37] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, R.R. Choudhury, No need to war-drive: unsupervised indoor localization, in: Proceedings of the 10th international conference on Mobile systems, applications, and services, ACM, 2012, pp. 197–210.

[38] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: BMVC 2009-British Machine Vision Conference, BMVA Press, 2009, pp. 124.1–124.11.

[39] J. Weickert, A. Bruhn, T. Brox, N. Papenberg, A survey on variational optic flow methods for small displacements, in: Mathematical Models for Registration and Applications to Medical Imaging, Springer, 2006, pp. 103–136.