Tool Support for Prototyping Interfaces for Vision-Based Indoor Navigation

Andreas Möller, Luis Roalter, Robert Huitl, Stefan Diewald
Technische Universität München
Munich, Germany
andreas.moeller@tum.de, roalter@tum.de, huitl@tum.de, stefan.diewald@tum.de

Christian Kray
University of Münster
Institute for Geoinformatics (ifgi)
Münster, Germany
c.kray@uni-muenster.de

Matthias Kranz
Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
Luleå, Sweden
matthias.kranz@ltu.se

ABSTRACT

Vision-based approaches are a promising method for indoor navigation, but prototyping and evaluating them poses several challenges. These include the effort of realizing the localization component, difficulties in simulating real-world behavior and the interaction between vision-based localization and the user interface. In this paper, we report on initial findings from the development of a tool to support this process. We identify key requirements for such a tool and use an example vision-based system to evaluate a first prototype of the tool.

Author Keywords

Indoor navigation; vision-based; augmented reality; virtual reality; mental models; prototyping

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

MOTIVATION AND BACKGROUND

Accurate, personalized and reliable indoor navigation is currently an area of research that is receiving a lot of attention[1] One of the key issues in this context relates to localization, as the standard outdoor method (GPS) is not available indoors. Therefore, a number of alternatives have been proposed, including Infrared beacons [1], WLAN/cell-based approaches [3, 8], and various sensor-based technologies [11]. More recently, methods for vision-based localization have been developed, e. g. using fiducial markers [10], or feature extraction [4] and image matching, which compares a query image (captured by the phone's camera) to a database of reference images with known location. Using the most similar reference image, the current position of the phone is estimated. Matching usually relies on characteristic portions of images, e. g. MSER (Maximally Stable Extremal Regions) [9] or text-related features [2]. The latter are particularly interesting for indoor environments, since signs and posters can be used as distinctive regions. In addition to estimating absolute locations, it is also possible to compute relative locations using visual odometry, e.g. by tracking features over time [7] or using inertial sensors to estimate the relative movement.

Vision-based indoor navigation is one promising candidate technology since it has the potential to work without an additional (costly) infrastructure, and since mobile devices are now capable to perform image recognition/analysis in real-time. However, since this approach is fairly new, it is still unclear how to best prototype, evaluate and compare different vision-based solutions and appropriate graphical user interfaces (e. g. Augmented Reality (AR) overlays, or map-based visualizations). Prototyping and evaluating such a system can be challenging for a number of reasons. Firstly, the effort for constructing an image database and the development of computer vision algorithms can be very large. Secondly, prototyping can be difficult as somehow the properties of the image recognition process have to be simulated (e. g. varying recognition rates). Thirdly, the performance of the computer vision component can interfere with the GUI. Consider for example an AR interface, where the localization is inaccurate and thus the AR overlay would be out of sync with the real world. Therefore, we need to more thoroughly investigate how vision-based navigation systems can be prototyped and evaluated.

In this paper we aim to address this knowledge gap by investigating how to support the prototyping and evaluation process for this type of applications. We first identify key requirements for tools supporting this process, and then describe the initial design and implementation of such a tool. We developed a novel vision-based indoor navigation interface, and used it to evaluate the proposed tool with respect to how well it meets the previously identified requirements and challenges. We discuss our observations and their implications for the evaluation and prototyping of vision-based indoor applications. The paper closes by summarizing our contribution and outlining possible future work.

REQUIREMENTS FOR PROTOTYPING TOOL SUPPORT

We deduce the requirements that need to be supported by our tool from the challenges of vision-based indoor localization.

Challenges

First, not all indoor locations are rich in unique visual features. Especially large buildings often exhibit sparse and ambiguous texture and areas resembling each other, such as e. g. hallways. Such route sections can partly be reconciled by odometry and dead reckoning, but only with continuously decreasing accuracy. As soon as a more distinctive region is in line of sight, it needs to be exploited to gain a new exact positioning. Active help of the user (directing the phone at such feature-rich regions) is required.

[1] http://support.google.com/gmm/bin/answer.py/answer=1685893

A second challenge is the inherent inaccuracy of the location estimate in form of localization and orientation errors (or a combination thereof), which can lead to misleading navigational instructions in the user interface.

Finally, we see the interaction concept for vision-based navigation systems as a central research question. To our knowledge, previous work focused to a large extent only on the technical foundations of indoor localization. However, it has to be investigated how user interface concepts support the image retrieval process and deal best with the particularities of vision-based navigation. For example, in order to acquire enough visual features, the user needs to hold the smartphone upright (as if taking a photo) from time to time, which cannot be taken for granted. Likewise, the users' intentions, goals and mental models must be taken into account.

Tool Requirements

We argue that a prototyping tool is necessary to address the above challenges. In particular, the system must support

1. the experimentation with and evaluation of concepts at an early point in the development process.
2. the evaluation at locations other than the intended deployment area, enabled by a quick setup for new locations.
3. the test and evaluation of different user interface concepts with regard to usability, efficiency and effectivity.
4. the simulation of inaccuracy of both the location and the orientation estimate, in order to find out how the user can deal with this inaccuracy using a particular visualization.
5. the evaluation of methods that induce users to point the smartphone to distinctive, feature-rich regions by appropriate affordances or motivational elements of gamification.

A TOOL FOR PROTOTYPING AND EVALUATING INDOOR NAVIGATION INTERFACES

Our evaluation tool is a Wizard-of-Oz prototype [6] and consists of each an Android app for the experimenter's (see Fig. 1, left) and the subject's smartphone (see Fig. 1, right).

Prototype Features

On the subject's (client) app, the user interface of a vision-based navigation system is modeled, independent of any actual localization implementation. Parameters such as direction instructions or triggers of certain user interface elements can be remote-controlled by the experimenter through the server app. In a later stage, the interface can also be linked with a live self-localization system.

The client application can render virtual reality scenes from images (see visualization concepts in the next section). Image sequences (each image typically taking around 30 kilobytes of storage) can either be received dynamically or stored locally for a predefined navigation path. In the latter case, just the position identifier needs to be received from the experimenter which reduces latency. The client app stitches received images to a panorama and projects them on a cylindric surface using OpenGL. A settings file specifies sets of images forming a panorama, the order of panoramas, and navigation directions to be shown on the client. Based on the received direction information, a 3D arrow is generated and rendered as overlay on the surface.

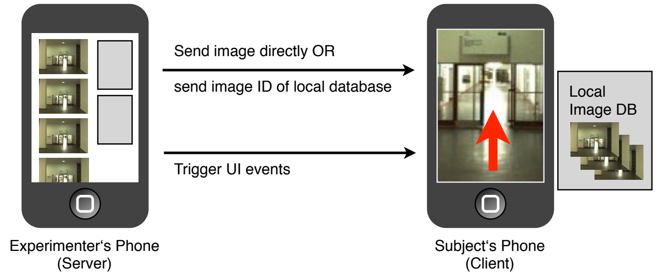

Figure 1. The experimenter's phone can send images for the virtual reality view and navigation instructions to the client, or trigger specific UI events and notifications. Image sequences can be cached on the client.

If offline image data is used, the client/server communication can be established using an ad-hoc network between the experimenter's and subject's smartphone, so that no WLAN infrastructure is needed in the testing environment. The live mode, where images and directions are received from the server, is extensible to be coupled with a 'real' localization service that provides matching panoramas (based on vision-based or other location sources).

Two Concepts for Indoor Navigation Visualization

To evaluate our tool, we designed a vision-based navigation application for a smartphone, incorporating two visualization concepts: augmented reality (AR) and virtual reality (VR) views. Both concepts share the idea of guiding the user with arrows (see Fig. 2).

Panorama-Based Virtual Reality

In this visualization, we use a virtual reality view based on a sequence of panorama images (comparable to Google Street View). Each panorama is composed of six images, which were previously acquired with a trolley-mounted Ladybug camera system[2]. We used the campus-wide indoor dataset [5], with a distance of 0.5 to 1 meters between subsequent panorama shots. In total, 5236 sets of six images forming a panorama were collected on a track length of 4522 meters.

Augmented Reality

The augmented reality view does not use panoramas, but shows the video stream of the smartphone's back camera. Navigation instructions such as walking directions are directly projected as overlay onto this live view.

Sensor Integration

The prototype uses the phone's compass sensor to automatically orient panoramas and navigation instructions. Alternatively, users can drag the panorama view around with their finger. Optionally, a 'rubber-band' effect can make the panorama snap back to the orientation provided by the compass when the user releases the view.

The phone's orientation sensor is used to switch between AR and VR view. When the phone is held upright, AR view is chosen. When the user is holding the phone with the camera

[2] www.ptgrey.com/products/ladybug3/ladybug3_360_video_camera.asp

pointing downwards, VR view is activated. The user can also manually switch between both modes.

Goal Distance Indicator

The system provides indicators to help the user estimate the distance to the goal. We propose different variants for such indicators (see right image in Fig. 2): showing the remaining time, the remaining distance or the remaining number of turns to the goal. Optionally, a progress bar can visualize the way to go in proportion to the distance already walked.

Figure 2. Left: Prototype showing a navigation arrow as overlay to the video image. Right: Variants for displaying the distance to the goal

Notifications

The quality of the location estimate relies on the number of available unique visual features. If the user carries the phone in a way that the camera cannot see sufficient features, the user is stimulated to target the phone at feature-rich areas. Those are typically found at eye height such as in posters on walls. We thus assume that the user will usually have to move up the phone along a vertical axis for a good position. In Fig. 3 we propose several visualizations to make the user perform that 'lifting' movement. Version *a)* uses focus for guiding to interesting areas (the more features the image contains, the sharper is the image). Version *b)* shows a text hint, in *c)*, an arrow on a color scale has to be directed to the green area, and *d)* uses a water level metaphor to find the correct height of the phone.

In case the user carries the phone in his hand (not currently looking at it) or in the pocket, the device vibrates to raise the user's attention (e.g. for feature indicators or for a turn instruction). In our prototype, all notifications can be triggered from the server application.

INITIAL EVALUATION

Review of Requirements

We review our prototype based on the above list of requirements (1–5). The system allows to test interfaces for visual navigation based on prepared image sequences using the Wizard of Oz technique. This enables quick and early prototyping and allows the usage of arbitrary image material for testing at any location (e.g. the research lab). We used the system to implement two user interfaces – virtual and augmented reality – which proves the suitability for comparing and evaluating alternative visualization concepts. Since the interfaces are implemented on real devices, their look and feel are close to 'real' systems, so that they qualify for evaluations of usability and effectivity. Requirements 1 to 3 are thereby met. Furthermore, arbitrary levels of inaccuracy can be simulated, since panorama images and navigation instructions are sent manually to the prototype. Finally, indicators encouraging users to point at feature-rich areas can be triggered by the experimenter, so that their effectivity can be measured. Requirements 4 and 5 are thus met as well.

Interface Design Review

In addition to a requirements-based review, we analyzed the tool based on the interface design proposals described above. Our goal was to assess the qualities of both approaches and to compare them without having to fully implement them. In prototyping, evaluating and testing these interfaces we were interested in the following research questions:

- Which visualization (AR or VR) deals better with the inherent inaccuracy related to position and orientation?
- Should the user interfaces be adjusted using the compass sensor or rotated manually?
- What are users' preferences for proactive behavior of the system, e. g. automatic adapation of visualizations based on the phone's pose and the users' goals and intentions?
- Are VR and AR interfaces sufficient for navigation tasks, or should an additional map be incorporated?

We conducted some informal test runs with 5 subjects in our university's main building with a predefined sequence of 91 panoramas over a distance of about 100 meters through corridors and a hall, including 4 doors and two turn instructions. The subjects used a Samsung Galaxy SII and the experimenter a Samsung Nexus S smartphone.

INITIAL RESULTS, DISCUSSION AND LIMITATIONS

Dealing with System Inaccuracy

Walking closely behind the test subject, the experimenter always chose the potentially best-matching panorama and sent it to the test subject. The resulting user experience of the directional arrow within the panoramas (AR) was considered as 'nice' and convenient. In AR view, the arrow's rotation based on the compass was not perfectly accurate, so that mismatches between the camera view and the on-top-projected navigation overlay occurred. The inaccuracy of the compass was perceived less disturbing in VR view: panoramas and navigation arrow also showed an offset in location and orientation, but they are coupled so that they are correct in relation to each other. We hypothesize that users can deal better with inaccuracy when using panoramas, since they can match the offset with the real world. Automatic rotation of panoramas seemed not even necessary; test subjects were also fine with moving them with their finger to orient themselves (the more as the correct initial orientation was set when the panorama was received from the experimenter). A challenge we came across was a noticeable lag of about one second for sending panoramas to the client, which complicated sending the correct panorama when the user was walking fast. The experimenter first tried to anticipate the correct image based on the user's walking speed and sent 1–2 images 'in advance'. Later we added a function for sending just every second or fourth panorama to keep up with the user's walking speed. This reduced step width of panorama updates (and only receiving a panorama update every few meters) did not seem problematic for users (it was maybe even advantageous since the interface became calmer). The ideal update rate will have to be investigated in the future.

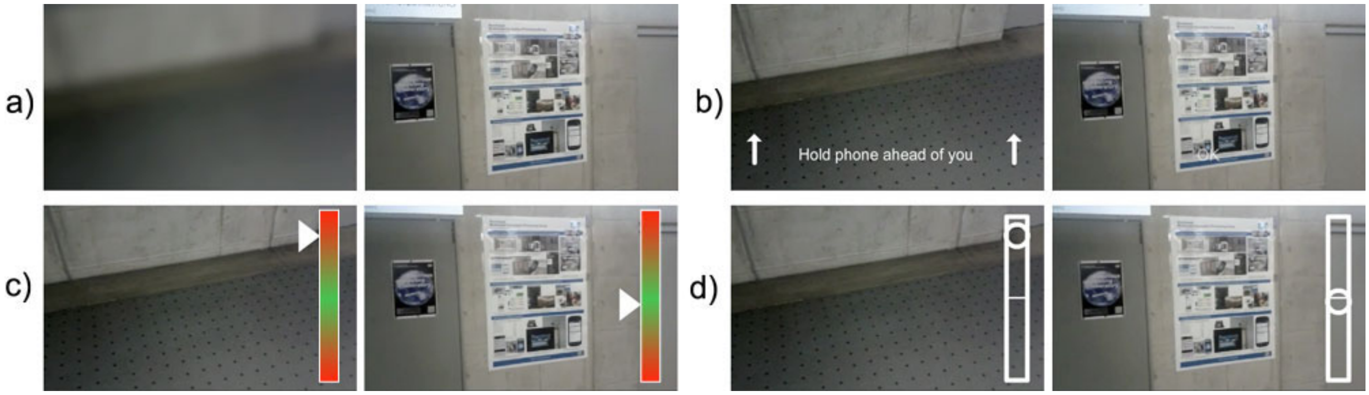

Figure 3. Proposed instructions to target the the phone at a feature-rich area: a) focus change, b) textual instruction, c) color scale, d) bubble level.

Proactive Behavior

We also experimented with context-sensitive switches between VR and AR based on the phone's position. Although the phone's position was reliably detected, automatic visualization changes were perceived as irritating (e.g. when a user raised the phone to match the panorama with his field of view, and the view automatically switched to AR). Here, the manual setting might be more appropriate.

The triggered notifications to direct the phone to a feature-richer area (as shown in Fig. 3) turned out to catch the eye; also vibration caused users to instantly look at the phone. Yet, we cannot evaluate with the present stage of the prototype whether the indicators actually ensured that sufficient visible features for location estimation are detected.

Map Requirements

In our prototype, no traditional map view was integrated and, at least for the small navigational task in this preliminary evaluation, not considered as missing. Future studies must reveal whether VR/AR views are sufficient for more extensive goal navigation and POI search tasks. Without digital technology, self-localization on a map was necessarily the first step of finding the way to a goal. However, this traditional mental model might be revised when appropriate alternative visualization concepts are used, making map localization as intermediate step unneeded.

CONCLUSION AND FUTURE WORK

In this work, we reported on the development and first experiences of a tool that supports prototyping and simulating the interaction between vision-based localization and the user interface. We presented initial findings on a vision-based navigation prototype we evaluated with our tool. In the future we will further extend our system and evaluate it qualitatively and quantitatively with a larger group of users.

ACKNOWLEDGMENTS

This research project has been supported by the space agency of the German Aerospace Center with funds from the Federal Ministry of Economics and Technology on the basis of a resolution of the German Bundestag under the reference 50NA1107.

REFERENCES

1. Butz, A., Baus, J., Krüger, A., and Lohse, M. A hybrid indoor navigation system. In *Proceedings of IUI 2001*, ACM Press (New York, 2001), 25–32.
2. Chen, H., Tsai, S., Schroth, G., Chen, D., Grzeszczuk, R., and Girod, B. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, IEEE (2011), 2609–2612.
3. Cheverst, K., Davies, N., Friday, A., and Efstratiou, C. Developing a context-aware electronic tourist guide: Some issues and experiences. In *Proceedings of CHI 2000* (Netherlands, 2000), 17–24.
4. Fritz, G., Seifert, C., and Paletta, L. A mobile vision system for urban detection with informative local descriptors. In *Computer Vision Systems, 2006 ICVS'06. IEEE International Conference on*, IEEE (2006), 30–30.
5. Huitl, R., Schroth, G., Hilsenbeck, S., Schweiger, F., and Steinbach, E. TUMindoor: an extensive image and point cloud dataset for visual indoor localization and mapping. In *IEEE International Conference on Image Processing (ICIP), Orlando, USA* (2012).
6. Kelley, J. F. An empirical methodology for writing user-friendly natural language computer applications. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '83, ACM (New York, NY, USA, 1983), 193–196.
7. Klein, G., and Murray, D. Parallel tracking and mapping on a camera phone. In *Proc. Eigth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'09)* (Orlando, October 2009).
8. Kranz, M., Fischer, C., and Schmidt, A. A comparative study of DECT and WLAN signals for indoor localization. In *8th Annual IEEE Int'l Conference on Pervasive Computing and Communications (PerCom 2010)*, IEEE (March 2010), 235–243.
9. Matas, J., Chum, O., Urban, M., and Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing 22*, 10 (2004), 761–767.
10. Mulloni, A., Wagner, D., Barakonyi, I., and Schmalstieg, D. Indoor positioning and navigation with camera phones. *Pervasive Computing, IEEE 8*, 2 (2009), 22–31.
11. Woodman, O., and Harle, R. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing*, UbiComp '08, ACM (New York, NY, USA, 2008), 114–123.