

# Interactive Positioning based on Object Visibility

Christian Kray<sup>1</sup> and Gerd Kortuem<sup>1</sup>

Computing Department, Lancaster University  
Lancaster, United Kingdom  
{kray,kortuem}@comp.lancs.ac.uk

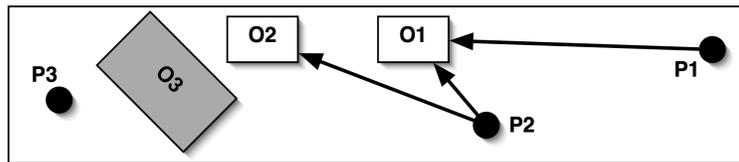
**Abstract.** In this paper we describe a new method and user interface for interactive positioning of a mobile device. The key element of this method is a question-answer style dialogue between system and user about the visibility of nearby objects and landmarks; answers given by the user provide clues about the relative position of the user and allow the verification or falsification of hypotheses about the user's absolute location. This new approach combines the respective strengths of a human user (i. e. fast and reliable object recognition) and a mobile system (i. e. fast computation of numerical data). It enables accurate positioning without requiring any other positioning technologies. A particular advantage of this approach is that it lends itself to the implementation on camera-equipped mobile phones, where it can be used to increase the accuracy of cell-based localisation methods.

## 1 Motivation

Location-based services (e. g. electronic tour guides [1], location-aware shopping assistance [2]) are one of the key application classes for mobile and handheld devices. Current methods for determining the location (position) of a mobile device (and thus its user) have a number of serious drawbacks in terms of reliability, accuracy, coverage and availability. For example, the quality of location information provided by GPS receivers is effected in unpredictable ways by external factors such as weather and nearby buildings. Essentially all current positioning technologies (including, for example, ultrasound indoor positioning systems, or positioning by detecting mobile phone network cells) have inherent limitations that effect the quality and reliability of the positional information. This fact makes the design of user interfaces for location-based services a challenging task. The basic question is how service fluctuations and disruptions due to technical limitations should be dealt with on the user interface level.

In this paper we propose that rather than trying to hide service fluctuations and disruptions from the user, the user can help to overcome them. We introduce a new interactive positioning method that involves a dialogue between system and user. This dialogue is driven by the system and requires the user to answer a few question regarding the visibility of prominent objects and landmarks (e. g. buildings). The answers provide clues about the relative position of a user with regard to these landmarks and can be used to determine the user's absolute position. The interactive positioning method combines the strengths of a human user – i. e. fast and reliable object recognition – and a mobile system – i. e. fast computation of numerical data – and provides a mean to determine the user's current position even in absence of any sensor readings.

Of course, asking the user to answer questions can be very disruptive and may lead to an unsatisfactory user experience. Thus crucial research issues related to interactive positioning are *when should the system initiate a dialogue* and *which questions should the system ask?* In this paper, we describe (1) how a dialogue can be generated from a set of initial hypotheses about the user's location (derived from possibly unreliable sensor observations) (2) how knowledge about topology and visibility constraints can be used to select appropriate landmarks and (3) how careful selection of landmarks can be used to minimise the number of questions to be asked.



**Fig. 1.** Varying visibility of objects from different positions: object O1 is visible from position P1 and P2, whereas O2 is only visible from P2 and neither O1 nor O2 are visible from P3

## 2 Interactive Positioning

The motivation for employing object visibility for an interactive positioning approach stems from the observation that during mobile phone conversations people often describe their current location, or inquire about the location of another person, in relation to prominent or well-known landmarks (e. g. “I can see a big church with a fountain next to it.” “Can you see a statue on a high pillar?”). Our approach takes the idea of dialogues about object visibility and moves it from the realm of computer-mediated human-human interaction to human-computer interaction. It is based on the following key concepts and assumptions:

- Objects (i. e. buildings, landmarks) are visible from a potentially infinite number of positions. Each position is a point in two- or three-dimensional space.
- Given a reference to an object (e. g. a verbal description or a photo), a user can determine whether or not the object is visible from their current position.
- A (finite) number of hypotheses about the user's current location can be derived either using traditional positioning systems or from initial estimates by the user. Each hypothesis refers to a single possible position.

Figure 1 illustrates these concepts. Three objects O1 to O3 are (partially) visible from positions P1 to P3: object O2 is visible from position P1 and P2, whereas O1 is only visible from P2 and neither O1 nor O2 are visible from P3. By asking questions about which objects the user can see (e. g. “Can you see objects O1 and O2?”) the system can infer whether the user is located at P1, P2 or P3. For example, if the user tells the

system that they can see O1, it can infer that the user is in fact located at P2 as this is the only position from which O1 is visible. However, learning that O3 is visible does not rule out any position hypothesis P1 to P3 as it is visible from all of them. It becomes apparent from this example that *for any given situation there is a very large number of questions regarding object visibility that the system could ask the user*, and that it is crucial for the effectiveness and user experience to select good object, i.e. objects which allow to eliminate many hypotheses once the system knows whether they are visible from the user's current position. This ability to select good objects is one of the key features of the algorithm we will present in the following section.

## 2.1 Integration with Existing Positioning Techniques

Most positioning technologies for mobile devices rely on sensors and direct measurements. Figure 2 provides a schematic overview of how interactive positioning can be combined with these. If sensor data is reliable and up-to-date, it provides a precise measurement of the user's current position and no further processing is required. If, however, sensor data is unreliable or outdated, the measurement has a low confidence value. In that case, the system can improve confidence by asking the user for explicit confirmation. For example, it can use a personalised you-are-here map containing well-known landmarks [3] to enable the user to verify the position. In case of no sensor data, we can resort to exploration [4] (i.e. by querying other knowledge sources or by asking the user for some rough estimation of their position such as "What quarter of the city are you in?"). While confirmation and exploration are a form of interactive positioning, our approach goes beyond simple dialogues by means of an optimised interaction with the user regarding the visibility of objects. The combination of direct measurement, inference and interactive positioning allows for the determination of the user's current position in a wide array of situations, where individual techniques would fail. In particular, our approach is even able to estimate the user's position without any sensor data at all (e.g. if the user provides some rough initial estimate such as "I am on a market square.").

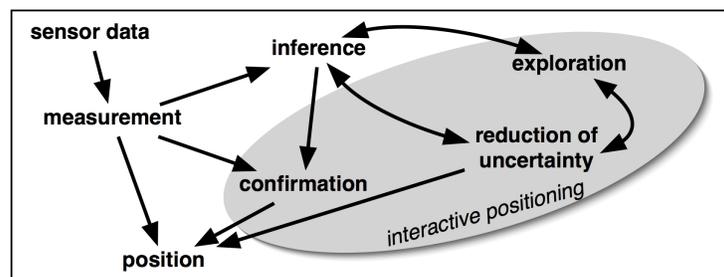


Fig. 2. Determination of the user's current position: an overview

## 2.2 Technical Requirements

In order to realise interactive positioning, a mobile system has to meet two key requirements. First, it needs to provide a user interface suitable for interactive dialogues between system and user. This may be done visually using a graphical user interface (GUI) or verbally using speech synthesis and speech recognition. Second, it needs to have access to a geographic information system (GIS). This GIS must contain a world model that allows for the computation of object visibility from arbitrary positions.

In the following section we describe how to generate a suitable dialogue from basic hypotheses. The discussion assumes a graphical user interface with the capability of displaying photos of individual objects and slideshows of sets of objects. A concrete example is given in Section 4.

## 3 Generating Dialogues for Interactive Positioning

Interactive positioning based on object visibility is an iterative process that takes as input a number of hypotheses (e. g. generated from imprecise measurements or through dead reckoning), and calculates a number of questions concerning the visibility of objects in the surroundings. These questions are optimised towards quickly determining the current position of the user. Figure 3 shows an overview of the entire algorithm that reduces the uncertainty, which position of a set of several candidates is most likely the actual position of the user. We first select the best divider, i. e. the salient object that partitions the visibility matrix (see below) in a way that allows us to quickly reduce the size of the matrix once we know whether the object is visible. In order to limit the number of interactions, we select the best dividers for the resulting sub-matrices as well. Then, we generate the query for the user, which consists of a repeating slide show of labelled images of the selected salient objects. The user's reply reduces the matrix according to the procedure described below. The algorithm terminates either when the user's position has been determined, or when it cannot identify it. In the latter case, we can resort to exploration (see Section 2).

In a first step, the algorithm retrieves all world objects that are close to the potential positions. In order to reduce the number of objects, a preselection based on their respective salience (see, for example, [5]) should be performed. For all the *salient objects* of the resulting set  $S$ , we then determine whether or not they are visible, i. e. for all potential positions we check the visibility of each object. We then dispose of a *visibility matrix*  $V(S, P)$  as shown in Figure 4, the central data structure for the algorithm.

Since the user's reply to a question – whether or not they can see an object – should allow us to eliminate as many hypotheses as possible, we have to select those salient objects that best partition the set of the potential positions. An ideal example for such an item would be a salient object that is visible from exactly half of the potential positions. Usually, there is no such object, and we have to instead select the ones that partition the set of potential positions in two sets of roughly the same size. More formally, we are looking for the salient object  $s_k$  (see Figure 4 for the definition of the terms used) for

```

select best dividers
    find salient object that best divides matrix
    find salient objects that best divide submatrices
generate query for user
    retrieve images for salient objects
    show repeating slide show and question
evaluate the user's reply
    reduce matrix
    if matrix is empty
        exploration
    else if matrix has only one element
        return as user's position
    else if elements of matrix can be merged
        return as user's position
    else if matrix does not allow for further reduction
        exploration
    else repeat

```

**Fig. 3.** The reduction algorithm: an overview

$$V(S, P) = \begin{pmatrix} vis(s_1, p_1) & \dots & vis(s_1, p_n) \\ vis(s_2, p_1) & \dots & vis(s_2, p_n) \\ \vdots & \ddots & \vdots \\ vis(s_m, p_1) & \dots & vis(s_m, p_n) \end{pmatrix}$$

$$S = \{s_i | 0 < i < m + 1\}$$

$$P = \{p_j | 0 < j < n + 1\}$$

$$vis(s_i, p_j) = \begin{cases} 1 & \text{iff } s_i \text{ is visible from } p_j \\ 0 & \text{otherwise} \end{cases}$$

**Fig. 4.** Visibility matrix and its constituents: the set  $S$  of salient objects  $s_i$ , the set  $P$  of all potential positions  $p_j$ , and the visibility function  $vis(s_i, p_j)$ .

which the following statement holds:

$$s_k : \forall i \in \{1, \dots, k-1, k+1, \dots, m\} : \left| \frac{n}{2} - \sum_{j=1}^n vis(s_i, p_j) \right| \leq \left| \frac{n}{2} - \sum_{j=1}^n vis(s_k, p_j) \right|$$

If more than one salient object meets this criterion we can either randomly select one, or recursively determine which salient object entails the lowest number of questions once its visibility is known. The latter alternative yields a more informed choice (at the expense of higher computational costs), since we analyse *in advance* what questions will follow when the user either confirms visual contact with the current salient object or not. This approach can also be iterated after each reply by re-evaluating the set of the remaining candidates in the same way (again at the expense of higher computational costs). Note that this approach ensures that the number of interactions required to determine the user's current position is minimised as it always selects those objects that result in the highest expected information gain.

Once the user provides the system with visibility information (either for one salient object or for several), we can adjust the visibility matrix by eliminating all positions that contradict the user's reply. The row(s) corresponding to the salient object(s) included in the query can be removed as well since it is of no further use. Figure 5 shows the formal procedure of elimination for a single salient object question. (Multiple salient objects questions can be treated as a sequence of single salient object questions.)

$$\begin{aligned}
 elim_s(k, V) &= \begin{pmatrix} vis(s_1, p_1) & vis(s_1, p_2) & \dots & vis(s_1, p_n) \\ \vdots & \vdots & \ddots & \vdots \\ vis(s_{k-1}, p_1) & vis(s_{k-1}, p_2) & \dots & vis(s_{k-1}, p_n) \\ vis(s_{k+1}, p_1) & vis(s_{k+1}, p_2) & \dots & vis(s_{k+1}, p_n) \\ \vdots & \vdots & \ddots & \vdots \\ vis(s_m, p_1) & vis(s_m, p_2) & \dots & vis(s_m, p_n) \end{pmatrix} \\
 elim_p(j, V) &= \begin{pmatrix} vis(s_1, p_1) & \dots & vis(s_1, p_{j-1}) & vis(s_1, p_{j+1}) & \dots & vis(s_1, p_n) \\ vis(s_2, p_1) & \dots & vis(s_2, p_{j-1}) & vis(s_2, p_{j+1}) & \dots & vis(s_2, p_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ vis(s_m, p_1) & \dots & vis(s_m, p_{j-1}) & vis(s_m, p_{j+1}) & \dots & vis(s_m, p_n) \end{pmatrix} \\
 V' &= elim_s(x, V_k) \text{ where } k = \sum_{j=0}^n vis(s_x, p_j) \\
 V_l &= \begin{cases} l = 0 : V \\ l \neq 0 : elim_p(\min\{j \mid j < columns(V_{l-1}) \wedge vis(s_x, p_j) = 0\}, V_{l-1}) \end{cases}
 \end{aligned}$$

**Fig. 5.** Elimination of false hypotheses in the visibility matrix

In order to determine the updated visibility matrix  $V'$  we need to eliminate all potential positions from the original matrix  $V$  from which salient object  $s_x$  is not visible. This

is an iterative process that removes position  $p_j$  if  $vis(s_x, p_j) = 0$  resulting in intermediate matrices  $V_l$ . Once all invisible positions have been eliminated, the current salient object  $s_x$  can be removed as well ( $elim_s(x, V_k)$ ). If the user reports that  $s_x$  is invisible, the only differences are that we have to eliminate positions  $p_j$  if  $vis(s_x, p_j) = 1$ , and that  $k = n - \sum_{j=0}^n vis(s_x, p_j)$ . If the user provides information about multiple salient objects simultaneously we can apply the same procedure to one salient object after the other. The process of interaction and elimination continues until

- **all hypotheses have been eliminated.**  
This implies that either the original set of potential positions was wrong, or that the user was unable to recognise a salient object, or has overlooked one or more salient objects.
- **there is only one hypothesis left in the visibility matrix.**  
We have successfully determined the user's current position, and the system can proceed with the task that requested positional information.
- **the remaining hypotheses can be merged into a single position.**  
This happens when the remaining salient objects do not allow for a reduction of uncertainty (e. g. they are visible from all positions), and if the remaining hypotheses are located close to each other.
- **the remaining hypotheses cannot be merged.**  
In this case, the remaining salient objects cannot be used to reduce the uncertainty, which of the remaining positions is the the true position of the user, and they are also too far apart to be merged.

The second and third case allow for the termination of the positioning process, and enable the system to continue to work on the task that originally requested information about the user's current position. In the first and fourth case, however, the reduction of uncertainty has failed, and other means have to be employed to still provide the service the user has asked for. A convenient possible 'by-product' of the interaction on the visibility of objects is a hypothesis about the current orientation of the user: in order to answer visibility questions, they mostly likely will look towards the objects in question and align themselves accordingly. A further beneficial side effect of the interaction is the introduction of a number of world objects that the system can later refer to, e. g. when generating localisations.

## 4 Case Study

We now describe a case study of using interactive positioning in a mobile tourist guide. We first illustrate the user experience when interacting with the prototype system and present some qualitative and quantitative results both from lab tests and a field trial.

### 4.1 User interface

Figure 6 shows a snapshot of the user interface of Deep Map, a mobile tourist guide that uses interactive positioning. Deep Map provides visitors of the city of Heidelberg with

a number of location-based services. The upper half of the figure depicts an example interaction: The system asks the user, which of three objects are visible from their current position. In order to facilitate recognition, a slideshow is presented to the user that consists of photographs of the objects that are annotated with their name. The slideshow is repeated until the user replies to the query. This does not only enable the user to identify the objects in their environment but also provides an easy mean to refer to them when replying to the system. The lower half of the picture illustrates the actual context of this interaction: the user is currently located on Seminarstraße and the objects mentioned in the query are nearby that street (all are highlighted in the map).

The interface shown in Figure 6 is one step in a (short) series of questions that occur during interactive positioning. Once the user provides the system with a reply – in our implementation via a pop-up menu and/or an on screen keyboard – it can update the visibility matrix and generate the next question if necessary. Once Deep Map has successfully determined the current position, it will update the internal position history and provide the user with its location-based services, e. g. personalised you-are-here maps such as shown in Figure 7.

## 4.2 Evaluation

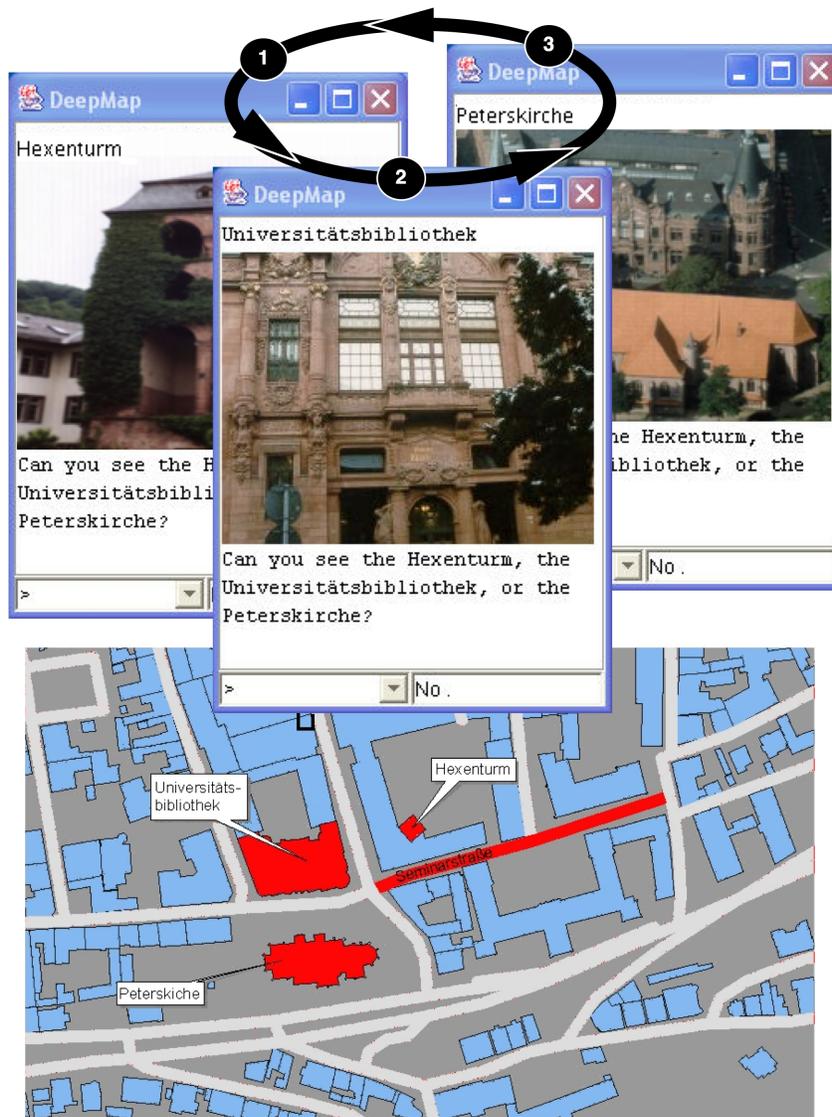
Our prototype is an extension of the Deep Map system [6], a system that provides services such as incremental navigation, information on sights, hotel reservation, and interactive maps. Deep Map relies on GPS to determine the user’s current position but has been designed to easily integrate other positioning techniques.

Deep Map was tested during development within the lab using a GPS simulator agent that allowed us to simulate accurate measurements as well as the complete lack of any readings. In those tests, we found the system to be able to determine the current position in a number of different situations, ranging from the complete absence of positional information to a set of different position hypotheses. However, on open places (such as market squares or wide roads), we observed a higher number of cases, where the system was not able to pinpoint the user’s position beyond a relatively low precision. We attribute this to the implementation of the visibility check: The algorithm used to compute visibility was based on a two-dimensional ray-tracing approach and therefore could not evaluate the visibility of objects that are further away but tall.

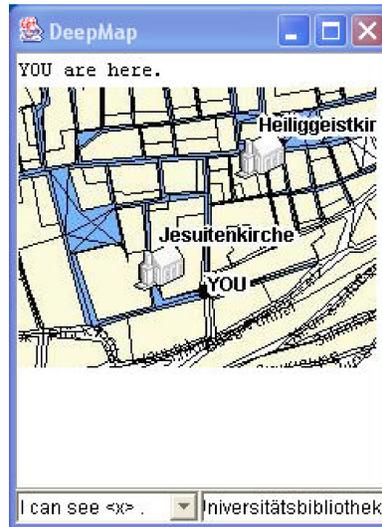
In addition to lab tests we also conducted a field trial with the system. In this case, we disabled the GPS while the user was on Seminarstraße (see also Figure 6) so that the system did not have any current sensor data. This street is approximately 170 meters long, and the prototypical implementation was able to determine the user’s current position in three interactions such as the ones shown in Figure 6. The computed position was accurate within an ten meter radius.

## 5 Related Work

There are a few systems – mainly prototypical tourist guides or navigational assistants – that already incorporate interactive positioning in one form or another. The GUIDE project [1] was developed at the University of Lancaster, and aims at providing visitors



**Fig. 6.** An example interaction: The user is located somewhere on Seminarstraße (highlighted on the map), and the system now asks whether ‘Hexenturm’ (1), ‘Universitätsbibliothek’ (2), or ‘Peterskirche’ (3) are visible (also highlighted on the map). Images of these three objects are shown in a continuously repeating slideshow (indicated by the circular arrows at the top of the figure) along with their name. In the prototype, the user could reply to the question in two ways: either by using the pop-up menu in the lower left hand corner of the screen/window to select a predefined answer (such as “Yes.” or “No.”) or by inputting free text into the input box in the lower right hand corner.



**Fig. 7.** A personalised you-are-here map that was generated after successfully applying interactive positioning (the objects shown on the map are known to the user).

of the city with information adapted to their interest and location. GUIDE can present its user with a list of all sights from which they can pick the one that is located nearest to them. Based on this selection, the system can then estimate the user's current position. While the list of sights GUIDE presents to the user is static, the LoL@ system [7] is able to dynamically generate it. LoL@ is also aimed at tourists, who it provides with information about points of interest, navigational assistance, and further location based services. Currently, it relies on GPS for positioning but it has been designed to exploit the position information provided by third generation mobile phones. In case LoL@ is unable to precisely determine the user's current position from sensor readings or through dead reckoning, it dynamically creates a list of street segments and ask the user to select the one they are located on. This list consists of ranges of house numbers along with the name of the street. Hence, this approach requires LoL@ to know the street the user is on.

A further interaction technique used for positioning consists of interactive maps, where the user can 'point' to their current location by clicking on the corresponding area on the screen of a PDA. Within the project REAL [8], for example, the imprecision of positional information is compensated by displaying a larger area of the environment. The user can then click on specific icons embedded in the map to tell the system about their current location. This information is then used to improve the quality of the presentation, i. e. by providing more precise route instructions. Bhasker et al. [9] use a similar approach to improve the precision of WLAN-based positioning but store user corrections as 'virtual access points' for later use.

However, there are several shortcomings in the approaches presented above. A static list of sights does not scale well – in a larger city, a user might have to select from thousands of items – and also restricts the precision of the resulting positional information. A dynamically generated list of street segments overcomes this problem to some degree but does require information about the street the user is in. In addition, longer streets will result in a long list of street segments, which are in turn hard to communicate to the user on a mobile device with limited screen estate. Interactive maps enable the user to quickly communicate their current location to the system but not only do they have to know their position rather precisely but they must also be able to indicate it on a map.<sup>1</sup>

The approach proposed in this paper addresses these issues in several ways. The iterative nature of the algorithm allows for a fine-grained control of the number of objects to present to the user. In addition, the objects are selected to maximise the expected information gain - hence, once their visibility is known the number of remaining alternatives is drastically reduced. Furthermore, contrary to interactive maps, our algorithm will work even if the user does not have any idea about their current location. It only expects the user to be able to visually scan their environment and to recognise objects that are presented to them.

## 6 Discussion

In our case study, we put the ‘burden’ of checking the visibility of objects on the human user but our approach would also support a system-side check of visibility. This opens up an interesting application area for mobile phones that are equipped with a camera. Instead of going through a number of interactions in order to determine their current position, a user could simply take a few snapshots of their actual environment and send them to a server. The server could then perform image analysis to match the photographs with others that are stored in its geo-referenced database. This does then provide the visibility information needed to reduce the visibility matrix according to the algorithm presented in Section 3. A further advantage of applying our approach to mobile phones with a camera is the fact that the current network cell of the phone provides an initial seed for constructing the visibility matrix. Consequently, the search space for the image analysis is also reduced to those images that are linked to the area of the current cell.

## 7 Conclusion and Outlook

Designing user interfaces for location-based services is a challenging task due to the inherent limitations of traditional positioning technologies in terms of reliability, accuracy and coverage. In this paper we proposed a new method and user interface for interactive positioning that is able to overcome these limitations. The interface uses a system-driven dialogue to resolve question regarding the visibility of prominent objects

---

<sup>1</sup> An alternative approach to ‘interactive positioning’ consists of adapting the interactions in the context of services to low-precision positional information instead of trying to pinpoint the user’s position more precisely (cf. e. g. [10]). However, there is a minimum precision for most services, which has to be met in order to provide them at all.

and landmarks; answers given by the user provide clues about the relative position of the user and allow for the verification or falsification of hypotheses about the user's absolute location.

In this paper, we described a method for generating a dialogue from basic hypotheses and we demonstrated how interactive positioning based on object visibility can be integrated into a mobile tourist guide system. Unlike previous approaches, our approach dynamically adapts the interaction to maximise the information gain from each interaction step while minimising the length of the interaction. The proposed mechanism not only allows one to specify how precisely the position of the user has to be determined but also seamlessly integrates with non-interactive approaches. A particular advantage of our approach is that it lends itself to an implementation on camera-equipped mobile phones where it can be used to increase the accuracy of cell-based localisation methods

## Acknowledgements

The work presented in this paper was funded by the Klaus Tschira Foundation, Heidelberg, Germany in the context of the project SISTO.

## References

1. Cheverst, K., Davies, N., Mitchell, K., Friday, A.: Experiences of developing and deploying a context-aware tourist guide: the GUIDE project. In: *Mobile Computing and Networking*, Boston, U.S.A. (2000) 20–31
2. Bohnenberger, T., Jameson, A., Krüger, A., Butz, A.: Location-Aware Shopping Assistance: Evaluation of a Decision-Theoretic Approach. In Goos, G., Hartmanis, J., van Leeuwen, J., eds.: *Proceedings of Mobile Human-Computer Interaction 2002*, New York, Heidelberg, Berlin, Springer (2002) 155–169
3. Sorrows, M.E., Hirtle, S.C.: The nature of landmarks in real and electronic spaces. In Freksa, C., Mark, D.M., eds.: *Spatial Information Theory (Proceedings of COSIT 99)*, Berlin, Heidelberg, New York, Springer (1999) 37–50
4. Kray, C.: *Situated Interaction on Spatial Topics*. PhD thesis, Computer Science Department, University of Saarland, Saarbrücken, Germany (2003)
5. Lynch, K.: *The Image of the City*. MIT Press, Cambridge, MA (1960)
6. Malaka, R., Zipf, A.: Deep Map - challenging IT research in the framework of a tourist information system. In Fesenmaier, D.R., Klein, S., Buhalis, D., eds.: *Information and communication technologies in tourism 2000*. Springer, Berlin, Heidelberg, New York (2000) 15–27
7. Anegg, H., Kunczler, H., Michlmayr, E., Pospischil, G., Umlauf, M.: LoL@: designing a location based UMTS application. *Elektrotechnik und Informationstechnik* **119** (2002) 48–51
8. Butz, A., Baus, J., Krüger, A., Lohse, M.: A hybrid indoor navigation system. In: *Proceedings of IUI 2001*, New York, ACM Press (2001) 25–32
9. Bhasker, E., Brown, S.W., Griswold, W.G.: Employing user feedback for fast, accurate, low-maintenance geolocationing. Technical Report CS2003-0765, UC San Diego, San Diego, CA, USA (2003)
10. Sood, S., Hammond, K.J., Birnbaum, L.: Low-fidelity location based information systems. In Nunes, N.J., Rich, C., eds.: *Proceedings of IUI 04*, New York, NY, USA (2004)