Landmark-Based Pedestrian Navigation with Enhanced Spatial Reasoning

Harlan Hile[1,2], Radek Grzeszczuk[2], Alan Liu[1,2],
Ramakrishna Vedantham[2], Jana Košecka[2,3], and Gaetano Borriello[1]

[1] University of Washington
[2] Nokia Research Center
[3] George Mason University

Abstract. Computer vision techniques can enhance landmark-based navigation by better utilizing online photo collections. We use spatial reasoning to compute camera poses, which are then registered to the world using GPS information extracted from the image tags. Computed camera pose is used to augment the images with navigational arrows that fit the environment. We develop a system to use high-level reasoning to influence the selection of landmarks along a navigation path, and lower-level reasoning to select appropriate images of those landmarks. We also utilize an image matching pipeline based on robust local descriptors to give users of the system the ability to capture an image and receive navigational instructions overlaid on their current context. These enhancements to our previous navigation system produce a more natural navigation plan and more understandable images in a fully automatic way.

1 Introduction

Mobile phones provide users with highly portable and connected computing devices. Moreover, the trend towards increased performance and inclusion of new sensors such as GPS and cameras in mobile phones make them a compelling platform for location-based services. In particular, navigation is emerging as a critical application for the mobile phone industry. We extend our previous work [1] on automatically generating landmark-based pedestrian navigation instructions with improvements on multiple fronts. In addition to improving landmark selection to provide more natural directions, we utilize computer vision techniques to improve both image selection and the quality of arrows augmenting the image. This extension also allows us to support the live annotation of images as the user follows a path.

Consider the situation of a visitor attending a talk on a university campus. A user can use their GPS enabled mobile device to navigate the campus by entering their desired destination. In the simplest case, this navigation aide may just be calculating a path and displaying it on a map along with the current GPS location. Matching the physical environment to the map may still be challenging, even if there are landmarks labeled on the map, as in Figure 1A. It is also

H. Tokuda et al. (Eds.): Pervasive 2009, LNCS 5538, pp. 59–76, 2009.

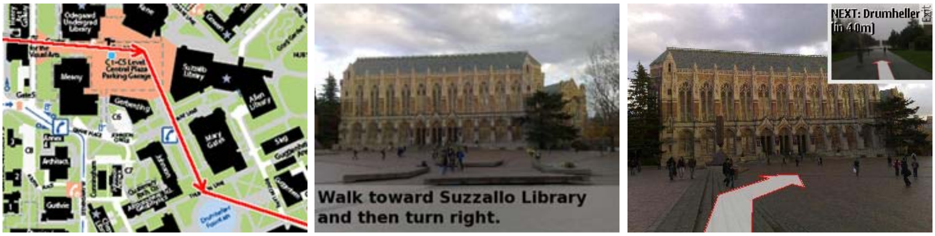


Fig. 1. Examples of landmark-based instructions. A (left): A map client with landmarks labeled. B (middle): Using text and canonical images C (right): Our system, utilizing reconstructed camera pose to accurately augment images.

possible to generate text-based instructions referencing landmarks, and provide accompanying images (Figure 1B). This makes it easier for a user to match to the physical environment, but without camera pose information for the images, an image may be chosen that is a significantly different perspective than the user sees. To lower the cognitive load further, we utilize the reconstructed camera pose to choose an image that is similar to the expected view of the user, and to automatically draw accurate arrows on the image, as in Figure 1C and Figure 2. This makes it easier for the user to orient themselves with respect to the images and the path. As a user walks along the path, the GPS location can be used to automatically show the next direction.

Many studies have shown that landmark-based navigation instructions provide significant benefits over map or distance-and-turn based directions [2,3,4]. Landmark-based navigation instructions are easier to follow, shorten the navigation time, and reduce confusion by providing visual feedback on the correctness of a navigation decision. Our previous work addressed the challenge of automatically creating landmark-based navigation instructions by leveraging an existing collection of geotagged images [1]. This work demonstrated the possibility to produce a set of navigation instructions utilizing these images. It also showed that users are able to follow these instructions and preferred them over other types of directions. The user studies from this system guided us toward the multiple improvements presented here.

Our previous system made decisions about which landmarks and images to use on a local basis. Here we improve on this by including higher-level reasoning to choose landmarks across larger regions of the path. This ties into user comments about the prior system indicating that text directions to accompany the images were important. While generating text corresponding to a single image is relatively straightforward, we aim to produce an entire set of directions that fits naturally with the way people navigate. For this reason, we have developed a set of heuristics to guide landmark choice. We also optimize landmark choice over larger sections of the path to provide a smooth flow. In addition, we provide support to fallback on map-based directions when appropriate landmarks or images are not available.

The previous system rendered arrows onto the chosen images using rough estimates for viewing direction and camera tilt. While this worked well in many

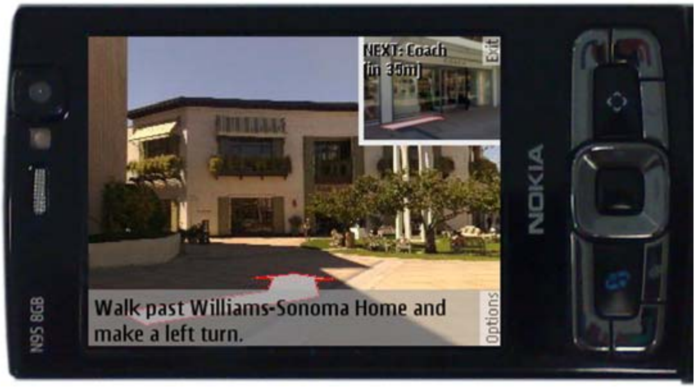

Walk past Williams-Sonoma Home and make a left turn.

Fig. 2. A sample client view of a generated instruction. The current direction is displayed prominently, but a preview of the next step is shown in the top right corner.

cases, it would occasionally produce confusing augmentations due to high GPS error or landmarks and camera poses that violated the standard assumptions. For example, see the uncorrected case in Figure 5. To address these problems we run an automated reconstruction algorithm to solve for full 3D camera poses of the images in the database, where a camera pose is described as a 3D location and a 3D orientation. This corrects for GPS error and provides accurate camera pose information, allowing us to improve low-level spatial reasoning. This information improves both image selection and augmentation, leading to more realistic arrows without any manual labeling. It also allows us to solve for camera pose of a new image, giving us the ability to augment a live image a user has just taken. Live augmentation is done by matching the image provided by the user with the images in our database and computing its pose from the poses of the matching images [5]. Matching is done using a mobile phone implementation of an image matching pipeline based on robust local descriptors [6]. This enhancement opens the possibility of rendering the image as part of a larger context. In summary, our proposed framework provides more compelling landmark-based navigation instructions in a fully automated way.

1.1 Prior Work

Requirement studies and user surveys have shown that landmarks are by far the most predominant navigation cues and should be used as the primary means of providing directions for pedestrians [7]. Goodman *et al.* [8] showed that landmarks are an effective navigation aid for mobile devices—they shorten the navigation time, reduce the risk of getting lost and help older people lower the mental and physical demand required to navigate. See our previous publications for more background on the advantages of using landmarks in navigation on mobile devices [1].

Recently mobile phones are becoming a popular platform for Augmented Reality type applications. Kähäri and Murphy [9] demonstrate such an application running on the newly released Nokia 6210 Navigator mobile phone. It uses an embedded 3D compass, 3D accelerometer, and assisted GPS unit for sensor-based pose estimation. Viewfinder images are augmented with information about the landmark at which the user is pointing the camera. However, sensors are often not accurate enough to allow for precise augmentation. A closed loop approach based on robust image matching supported by our system alleviates this problem. Recently, Google has announced Street View for mobile [10]. The software downloads a panoramic view of the current location and supports walking directions. Unfortunately, it is being reported that data transfer speeds are limiting the usefulness of that system.

Efforts to build a guidance system based on online collections of geotagged photos include the work by Beeharee and Steed [2]. They built a prototype database of geotagged photos by manually entering the location and the viewing direction of each photo. The study found that with landmarks the users finished the route significantly faster than without them, and that users found the landmarks helpful and informative. Since the authors did not use any augmentation of images, the users found some of photographs that were not taken exactly along the navigation path confusing.

We previously extended their work by making landmark images of primary importance in generating directions and automatically augmenting images with navigation directions, which greatly increases the confidence of the users and their understanding of the navigation instructions [1]. The technique of leveraging collections of photographs is a good approach for mobile devices since it is lightweight and does not require special hardware, which are drawbacks of some other systems [11,4]. While this previous work was designed to choose landmarks with important features (good *advance visibility* and *saliency* [12]), it would occasionally result in confusing images due to a lack of quality orientation information. This was partially alleviated by manually labeling a subset of images with orientation information, but it unfortunately limited the number of images available for annotation. The previous approach also reduced landmark choice to a local decision, resulting in a set of instructions that was not very natural for the user.

1.2 Contributions

This work extends pedestrian landmark-based navigation into a number of new areas:

- We propose a set of heuristics to optimize landmark choice over a larger area in order to produce a more comprehensible set of instructions.
- We present an approach to automatically solve for camera orientation and correct poor GPS readings by leveraging computer vision techniques. Our system automatically reconstructs camera pose by using the Photo Tourism system [13]. In addition to providing orientation information for more pictures than the previous approach of manually labeling a subset of images, it also provides complete camera pose information with refined location. The

reconstruction step also serves as a filter for quality images, removing images with poor exposure, excessive clutter, or mislabeling. We extend Photo Tourism's computation by automatically aligning the reconstructed geometry to the world by using (possibly noisy) GPS data associated with the images.

- We support automatic live augmentation of images taken by the user, which was deemed important in the user feedback on our previous system. Instead of using a system based on matching to a 3-D reconstruction as is done in Photo Tourism and other previous work [13,14], we propose a new method based on image-to-image matching that can work with a system designed to run in real-time on a camera phone [6].
- Additionally, since our technique automatically aligns the 3D reconstruction of a landmark with the world data, the system can be further extended to support other features that users asked for in our initial study: features such as zooming out to give users additional context information (for example street information), warping images so that they appear as if they were taken from the current location, highlighting a portion of an image that contains the landmark, etc.

2 System Overview

This section describes the details of our landmark-based pedestrian navigation system with enhanced spatial reasoning. Figure 3 shows the block diagram of the system. Our system supports two types of user interaction: following a set of cached navigational instructions and augmenting a live image taken while the user is en route. Our navigational instructions are built using images from a database of geocoded and labeled images, as described in Section 2.1. In order to produce a set of navigational instructions from an input path, we first apply high-level spatial reasoning to optimize landmark selection over longer sections

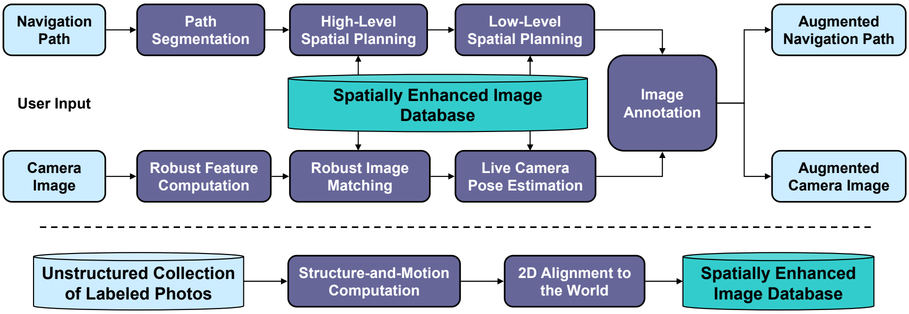

Navigation Path
Path Segmentation
High-Level Spatial Planning
Low-Level Spatial Planning
Augmented Navigation Path
User Input
Spatially Enhanced Image Database
Image Annotation
Camera Image
Robust Feature Computation
Robust Image Matching
Live Camera Pose Estimation
Augmented Camera Image
Unstructured Collection of Labeled Photos
Structure-and-Motion Computation
2D Alignment to the World
Spatially Enhanced Image Database

Fig. 3. Block diagram for landmark-based navigation supporting two modes: one that produces a set of augmented images for a user-supplied navigation path and one that augments a user-supplied image. Below the dotted line we show construction of the spatially enhanced image database from an unstructured collection of labeled photos.

of the path (Section 2.2). Next, we apply lower level spatial reasoning to optimize image selection locally and augment the images, as described in Section 2.3. In the case of the live image, we use a robust image matching pipeline adapted to mobile phones to find matching images in our database. We then use the poses of those images to compute the pose of the live image and augment it directly on the device. This is discussed in Section 2.4. Finally, in Sections 2.5 and 2.6 we describe how the unstructured database is processed to produce the spatially enhanced image database that includes 3D reconstruction of the landmark shapes and 3D camera poses for the images aligned to the real world. In each section we also present the results of each step.

2.1 Image Database Organization

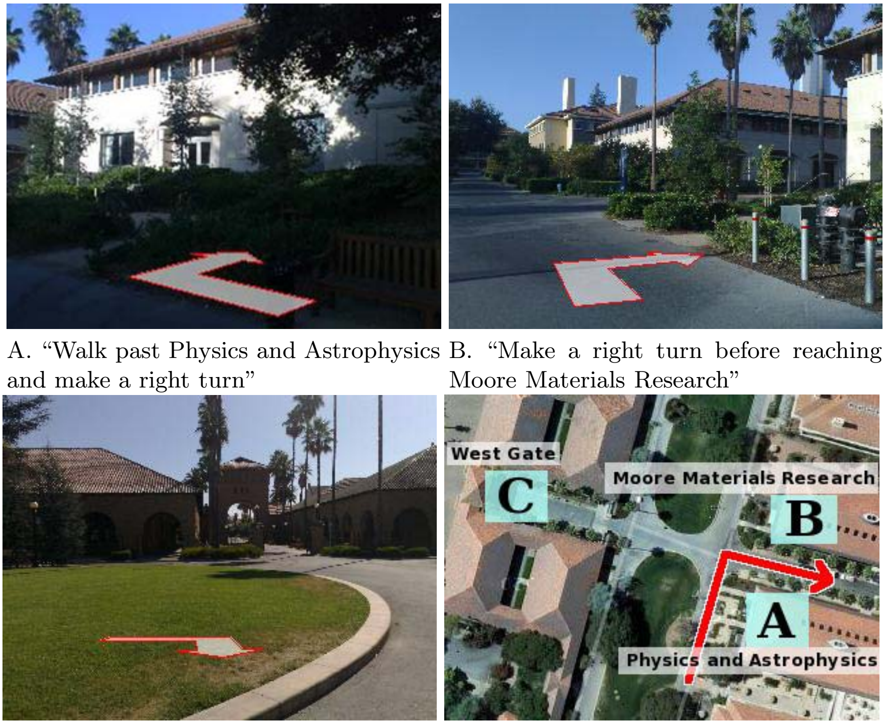
We are currently using a database of landmarks and images from an existing outdoor augmented reality project [6][1]. It was populated by many users taking pictures with GPS-enabled camera phones over a period of time. Each image was tagged during capture with the names selected from a list of nearby landmarks. Additionally, most (but not all) of the images were tagged with GPS location. The phones used in collecting the data did not have a built-in compass, hence no orientation information was recorded. In addition, the GPS accuracy is also limited, ranging between 10-100m. We choose to use this data instead of data from Flickr or other photo sharing services because the image tagging found on those sites is generally of poor or inconsistent quality.

The landmarks stored in the database also have an associated GPS location—a single point placed somewhere within the geometric extent of the landmark. This data may come from a mapping service or may be manually entered. As was shown in the past, this crude approximation to landmark location is often not sufficient for estimating accurate camera orientation, which is critical for a correct image augmentation [1]. To deal with this problem, we utilize additional information: a 3D camera pose calculated using computer vision algorithms. The camera pose is defined as a 3D location described in terms of longitude, latitude and altitude and a 3D orientation defined as a unit sphere vector. The details of how this information is computed is left until Sections 2.5 and 2.6, as we first discuss the applications of the spatially enhanced image database.

2.2 High-Level Spatial Reasoning: Selecting Landmarks for Natural Navigation

Navigation studies have shown that landmarks are important for pedestrian way-finding. Most importantly, landmarks are used to identify points where there is a change of direction [15]. In addition to identifying turns, they are also used to confirm travel in the correct direction [7]. Our system is given a path consisting of a series of GPS coordinates as input and outputs a complete set of navigation

[1] The readers can access our databases through a web-based interface available from this URI: `http://mar1.tnt.nokiaip.net/marwebapi/apiindex`.



A. "Walk past Physics and Astrophysics and make a right turn"

B. "Make a right turn before reaching Moore Materials Research"

C. "Make a right turn, West Gate will be on your left."

D. Map view of the same instruction, showing landmarks

Fig. 4. Example showing degrading choices of landmark at a corner. *A* uses a landmark inside the corner, producing a natural description. *B* uses a landmark on the inside half. *C* uses a landmark on the opposite side of the turn and is less optimal. *D* shows an option to show a map of the turn, with landmarks used in *A*,*B* and *C* labeled.

instructions. The previous system made local decisions about landmark choices based on visibility and saliency. We aim to incorporate the ideas of how people naturally navigate by choosing landmarks based on larger regions of the path. We have developed a set of heuristics that we believe will support this.

Our heuristics focus first on turns, since these are the important decision points along a path. Figure 4 shows the landmarks and images chosen for a sample corner using various options. When navigating a turn, it is most natural to reference a landmark on the inside corner of that turn, for example, "Walk past *landmark* and turn" (Fig. 4A). To achieve this, we look for landmarks in the inside quadrant of the turn that have images taken along the path approaching the turn (to measure visibility). If a landmark is not available in this region, we search the inside half of the turn ("Turn before *landmark*," Fig. 4B), and lastly fall back to landmarks on the outside of the turn (Fig. 4C). If there are still no appropriate landmarks, we can produce a map representation of the turn (Fig. 4D). If a landmark exists but has no appropriate images, it can still be referenced in the text directions and labeled on the map.

We also note that a turn consists of two parts: the path before the turn and the path after the turn. The view before the turn gives an indication of where to turn, and the view after the turn serves to confirm that the correct turn was made. For this reason, the navigation client shows the current step and the next step, so the user knows what to expect next, as seen in Figure 2.

When choosing landmarks for straight segments, it is important to choose landmarks with good visibility. However, choosing the landmark with the best visibility at each point along a route can result in a user seeing several different landmarks along a straight segment, even if a single landmark might be visible throughout the entire length. It is preferable to minimize this switching between landmarks in order to provide a more coherent navigation experience. To accomplish this, we define a cost function over the set of landmarks used at each point in a straight segment. We assign a small penalty for using less visible landmarks (in proportion to its visibility rank) and a large penalty for switching landmarks (adjustable to the desired level of landmark stability). We then find the optimal (least costly) set of landmarks using dynamic programming. We believe this produces more natural directions that allow users to navigate using landmarks as waypoints, rather than present a navigation experience that consists of precisely following a series of "micro-steps." Reducing the number of landmarks associated with a route also increases each landmark's significance and can promote learning the path. Although we believe these heuristics to align with desired properties in navigational instructions, we plan to carry out user studies to evaluate their effectiveness.

2.3 Low-Level Spatial Reasoning: Using Reconstruction for Image Selection and Augmentation

The previous step in planning the navigation instructions only selects which landmarks to use at different portions of the path. The next step is to select an appropriate image of that landmark at each location. This is accomplished by using the reconstructed 3D camera poses stored in the database. The computer vision reconstruction also serves as a filter for quality images: images with poor exposure, excessive clutter, or mislabeling are not likely to be reconstructed. This inherent filtering allows us to simply pick a reconstructed image that is close to the path and well aligned with the path. This is an improvement over our previous approach which only selected from a small set of images with high computed saliency, some of which were manually tagged with camera direction information. Having a larger set of images to choose from increases the likelihood of finding a good match to the current path.

Once an image is chosen, the reconstructed camera information can be used to augment the image with navigational instructions. Figure 5 shows an example of how this information is used to improve the quality of image augmentation. Without reconstruction, the camera orientation is estimated by simply using the direction from the GPS location of the camera to the GPS location of the landmark. This will produce poor results when the landmarks are large, when an image capture location is very close to a landmark, or when there is high GPS

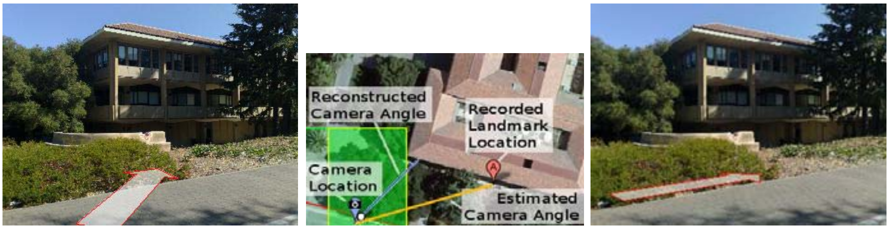


Fig. 5. An example of correcting image augmentation using reconstruction data. The goal is to specify a path which passes alongside the building. Compare the uncorrected case (left), which relies on GPS location of the camera and the landmark, to the case which uses the automatically computed camera pose (right). The center map shows the difference in camera angle between the two techniques.

error. This case can be seen in the left image of Figure 5, with the estimated camera direction shown in the center map. In addition to providing a more accurate orientation and corrected GPS coordinates, the camera pose also includes an estimate of tilt. This allows the arrow to be drawn in the correct orientation and also the correct perspective, making it seem better integrated with the image. The resulting image is shown in the right image of Figure 5. Another benefit of having full camera pose information is it opens the possibility of rendering the image in a new view with additional context, as discussed in future work.

2.4 Pose Estimation and Annotation of Live Images

We use the image matching pipeline for mobile phones developed by Takacs *et al.* [6] to support annotation of live images. First we compute the camera pose of the user-provided image from the poses of the matched images stored in our spatially enhanced image database. Next we use the computed camera pose to augment the live image using the same methodology as was described in the previous section for annotation of images from the database. An alternative approach is to try to register the user supplied image with the reconstructed 3D geometry using the structure-and-motion computation discussed in the next section. However, the approach we have chosen allows us to compute the camera pose and annotate the live image directly on the handset, thus reducing latency, bandwidth and computation. The whole process of finding matching images, computing the camera pose and augmenting the live image takes less than 3 seconds on a typical smart phone available today.

Figure 6 shows an example of a live image (shown in the middle) annotated using the camera pose computed from the poses of the two best matching images found in the database (shown on the sides). Given a new query view, we match it to the database using the existing system described by Takacs *et al.* [6]. From the returned images we select the top k images with the largest amount of geometrically consistent matches. Out of these we select the top two views (denoted (R_1, t_1) and (R_2, t_2)) which have been successfully registered in the stage described in Section 2.5 and with locations and orientations in the global world

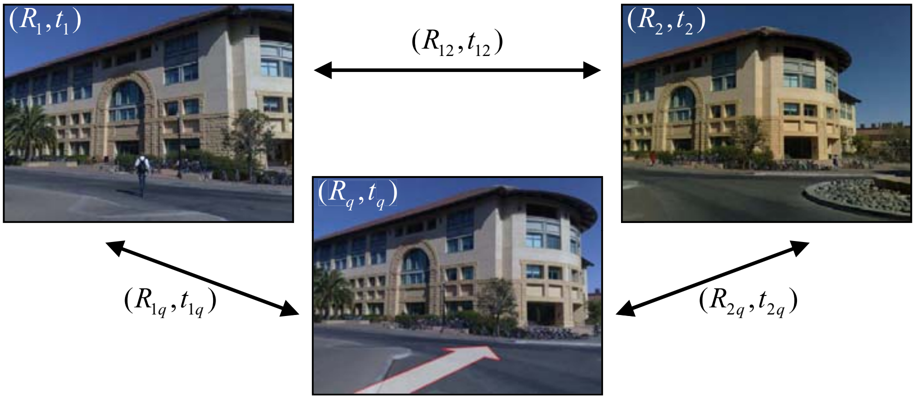


Fig. 6. An example of a live image (shown in the middle) annotated using the camera pose computed from the poses of the two best matching images found in the database (shown on the sides). The camera pose computed using this lightweight technique closely matches the result of the more complex structure-from-motion computation.

coordinate frame derived using the global pose alignment process described in Section 2.6. We denote the relative displacement between these two reference views to be (R_{12}, t_{12}). The position and the orientation of the query view can then be determined by triangulation using the following algorithm. Using the information about the focal length of the query view from the Exif (Exchangable image file format) tag stored in each image, we can compute the essential matrix E and consequently the motion between the query view and the two reference views. Let (R_{1q}, t_{1q}) be the motion between the first reference view and the query view and (R_{2q}, t_{2q}) be the motion between the second reference view and the query view, with both translations computed only up to scale. Note t_{12} and t_{1q} are with respect to the coordinate system of the first view, while t_{2q} is not. For the triangulation to proceed, we need t_{2q} in the coordinate frame of the first reference view $\tilde{t}_{2q} = R_{12}^T t_{2q}$. All three translation vectors are then projected to the ground plane. The three translation vectors in the coordinate system of the first reference view form a triangle. Knowing the absolute scale of t_{12} and the two angles between t_{12} and t_{1q} and t_{2q} we can use simple trigonometry to compute the correct location of the query view. The remaining orientation of the query view in the world coordinate frame is then $R_q = R_1 R_{1q}$.

We tested our pose estimation algorithm by comparing it to the camera pose computed using the structure-and-motion algorithm described below. For each query image we tested, the camera pose computed using our algorithm was nearly identical to the camera pose obtained using the more complex computation.

2.5 Structure-and-Motion Reconstruction

The navigation system enhancements detailed above require a way of automatically computing spatial image details from an unstructured set of images. We

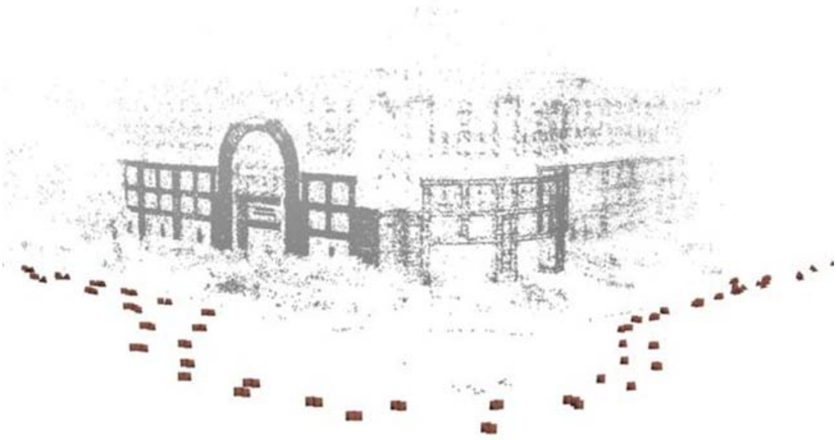


Fig. 7. Structure-and-motion reconstruction results for one of the landmarks in the database. 3D structure shown as a gray point cloud in the back and the reconstructed camera poses shown in red in the front.

propose an algorithm for computing the camera orientations and propagating other meta-data, such as GPS location, to the images in the database that do not contain this information. Additionally, we show that when meta-data such as GPS information or compass orientation exists, we can correct for the error in sensors. This can be accomplished by means of full 3D registration of available views using visual information only, thanks to robust large scale wide base-line matching using scale invariant image features and a structure-and-motion estimation algorithm. At this step we use an open source package developed by Snavely *et al.* [13] which facilitates fully automatic matching and full 3D registration of overlapping views. The core of the incremental and final pose registration algorithm is done by a modified version of the sparse bundle adjustment package of Lourakis and Argyros [16][2].

The structure-and-motion pipeline is used to improve the quality of the meta-data associated with the images in a collection of user contributed photos. Processing is done independently for each landmark. First, we extract from the database all images labeled with the given landmark. The camera registration pipeline detects SIFT features [17] in all the images. Features are matched between images and the resulting matches are pruned by enforcing geometric consistency.

Geometrically consistent views are incrementally registered together using the bundle adjustment algorithm after selecting an initial starting image pair. For more information on this algorithm, refer to the Photo Tourism paper [13]. This algorithm requires significant computation and can take hours to run. However, this is an offline process and the results are easily cached for later use. The

[2] The software is available from this URI:
`http://phototour.cs.washington.edu/bundler`

resulting reconstructed 3D points $\mathbf{X}_j$ and the registered camera poses (R_i, t_i) are given in the reference frame of the initial camera pair.

Figure 7 shows a result of structure-and-motion computation for one of the landmarks in the database. This reconstruction is one of the most detailed, since it was computed from 150 images. Typically our reconstructions are done using between 5 and 15 images, resulting in much sparser 3D structure. However, since we are mainly interested in the camera poses, which are always reconstructed very accurately, this is not an issue. Figure 8 shows results of aligning the reconstructed camera poses to the world.

Often the images representing a landmark will form disconnected clusters (images in separate clusters will not have any features in common). This is particularly common when we are dealing with large landmarks visible from dissimilar viewpoints. Due to this disconnection, it is often not possible to register all images of a landmark in a single stage. Instead of dealing explicitly with clustering of views prior to reconstruction, we deal with this issue by running the bundler registration incrementally. In the first stage we register as many images as possible and reconstruct the first cluster. We then rerun the reconstruction pipeline on the images which were not successfully registered in the previous stage and repeat these steps until no more poses can be reconstructed successfully. Although computationally not optimal, this naturally enables us to keep initializing the registration process with new views, which may have no overlap with the starting image pair. This process typically converges after at most three iterations.

2.6 Aligning 3D Reconstructions to the World

In order to utilize the computed camera pose information, the reconstruction must be aligned to the real world. The structure-and-motion algorithm result is expressed in the reference frame of the first selected image pair, and it is ambiguous up to the similarity transformation comprised of rotation, translation and scale. In order to align these results to the world, we use available GPS information extracted from the image tags. We use the grid-based UTM (Universal Transverse Mercator) coordinate system, since it makes alignment to the metric 3D reconstructions simpler than the latitude/longitude coordinates.

First we compute the gravity to ensure that the y-axis of all camera coordinate frames is perpendicular to the ground plane of the world. Using the formulation described by Szeliski [18], we estimate a global rotation of the entire reconstruction, which minimizes the deviation of the perpendicularity for all camera coordinate frames. All the camera poses can then be projected to the ground plane, where the 2D similarity transformation is then estimated. Since some of the GPS coordinates of the reconstructed images have large errors, we proceed to estimate the 2D similarity transformation $T_s = (R_s, t_s, s)$ in a robust way similar in spirit to the RANSAC algorithm.

The minimal number of poses and corresponding GPS locations needed to estimate the 2D similarity transformation is two. Given two reconstructed camera locations Cp_i and Cp_j and two corresponding GPS sensor readings L_i and L_j,

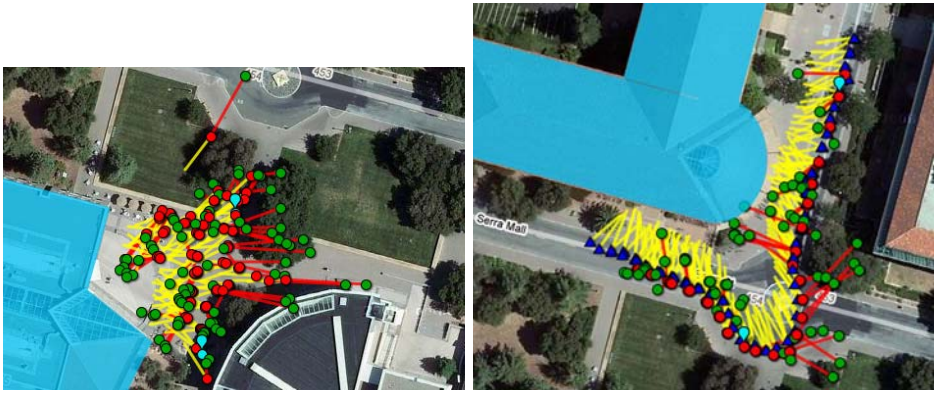


Fig. 8. Two examples of a pose alignment using our algorithm. High GPS sensor error is recorded in many cases indicated by long red lines connecting green (GPS reading) and red circles (reconstructed position). Yellow lines indicate reconstructed camera direction. Dark blue triangles are synthesized GPS location for images that did not have a GPS reading. The right image shows an alignment computed from the 3D reconstruction shown in Fig. 7.

the scale s can be estimated as the ratio of the two distances $s = \frac{\|L_i - L_j\|}{\|Cp_i - Cp_j\|}$. The translation t_s is then taken as the displacement vector between two mean locations of the chosen image pair $t_s = \bar{C}_p - \bar{L}$, and the rotation angle α_s is the angle between two lines connecting the reference GPS locations and the reconstructed image locations after the translation alignment t_s.

With a few GPS locations that are well distributed (not all close to each other), we can find a good alignment. From the set of n GPS locations we pick randomly 2 GPS locations that are relatively far away from each other and use them to compute a translation, rotation and scale hypothesis. Given the obtained hypothesis, we transform all reconstructed camera poses to obtain their GPS positions. We then evaluate the hypothesis by computing the total alignment error, which is the sum of distances between the original GPS location and the reconstructed GPS location for each camera pose for which GPS information was available. We repeat this hypothesis selection process k times and select the hypothesis which generated the smallest total alignment error. This optimization allows us to correct for errors in GPS positioning.

Figure 8 shows the results of this algorithm for two examples. The green dots indicate the original GPS locations and the red dots indicate the reconstructed camera poses. The original and the reconstructed GPS locations are connected by red line and the the yellow lines show the reconstructed camera orientations. The two cyan pins indicate the two GPS locations that were used to compute the similarity transformation.

It is worth noting that many original GPS locations have a high error (green dots showing on top of a building, or in the middle of a street). Relatively high error is characteristic of the GPS sensors in today's mobile devices. Our algorithm is able to identify this error and correct for it. The reconstructed poses tend to be very accurate, to the point that they can be used as the reference locations. The reconstructed orientations are also computed appropriately, since they are all pointing in the direction of the building facade.

It is difficult to determine the accuracy of this algorithm because ground truth information is not easily obtainable. In order to evaluate the reconstruction and alignment, we performed the following experiment. For a number of reconstructions, we confirmed by visual inspection that the reconstructed camera poses project to the locations where the images were taken from. We then computed the mean/max GPS reprojection error. The average correction across our examples is approximately 6 meters, with a maximum of 47 meters. For a number of images for which we had correct GPS locations, we manually labeled the approximate orientation. We then compared that orientation with the orientation we got from the reconstruction. The typical error was small, on the order of 10 degrees, and well within the error of our manual labeling. We also calculated how much the reconstruction estimate of angle changed over the previous estimate that relied on GPS location of the camera and landmark. For our examples, the average angle correction was 38 degrees, and the maximum observed was 170. This indicates the reconstruction has significant impact on the generated instructions.

An added benefit of computing the structure-and-motion reconstruction is that after the alignment, we can synthesize the GPS locations and the world camera orientations for the images that did not have this information originally. The reconstructed GPS poses for the images with no GPS sensor reading are shown as dark blue triangles in the right image of Figure 8. We can see by visual inspection that these positions are also reconstructed with good accuracy.

Alternative Alignment Method. There are situations where the above described algorithm fails; for example, when all images for a single landmark are

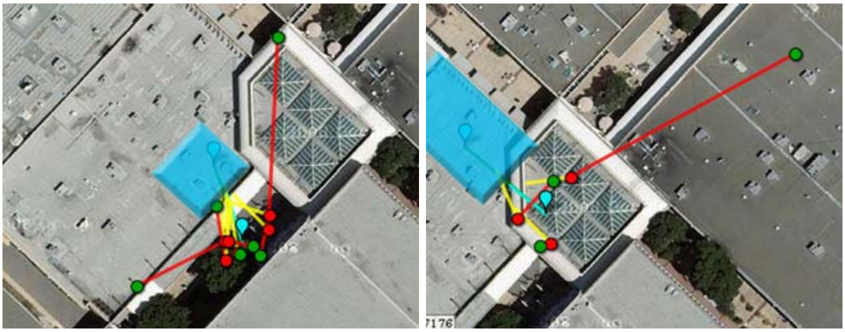


Fig. 9. Two examples of pose alignment using our alternative method for small landmarks. Some GPS locations have very large errors–on the order of a hundred meters. Meanwhile, the reconstructed poses correctly identify that all images were taken from roughly the same location. The orientation and scale are also correctly estimated.

taken from roughly the same location. This may happen when we are dealing with small landmarks, or landmarks with a single interesting feature. In this case, the GPS error starts to dominate the distances between the different GPS samples, which prevents us from having a robust rotation alignment. In those situations, we use a different technique for aligning the poses.

We first compute the mean GPS location $\bar{L}$ and the mean reconstructed camera location $\bar{C}_p$. We then search for those GPS locations that are far away from $\bar{L}$ (the distance is bigger than the mean distance) and consider those locations to be the outliers. We re-estimate $\bar{L}$ and $\bar{C}_p$ using only the inliers. As before, the translation t_s is taken as the displacement vector between the two mean locations: $t_s = \bar{C}_p - \bar{L}$.

We compute the mean orientation by averaging the direction computed from the GPS locations of the inliers to the landmark location (since this algorithm is used only for small landmarks, a single point approximation of the landmark location works quite well). We also compute a mean orientation of the camera viewing direction obtained from the reconstruction. We use the difference in those two directions to determine the alignment rotation R_s. The scale s is determined by computing the ratio of the mean distance to $\bar{L}$ and the mean distance to $\bar{C}_p$. Once we have the proper alignment, we can propagate the correct pose reconstruction to those cameras that were considered to be outliers (or those that do not have the GPS information available).

Figure 9 shows two examples. The green dots indicate the original GPS locations and the red dots represent the reconstructed GPS locations. The yellow lines indicate the reconstructed viewing directions. The two cyan pins connected by a line correspond to the landmark location and the mean GPS location $\bar{L}$. The direction of the line connecting them was used to estimate the rotation angle of the alignment. As we can see, some of the GPS locations have very large errors—on the order of a hundred meters. Meanwhile, the reconstructed poses correctly identify that all images were taken from roughly the same location.

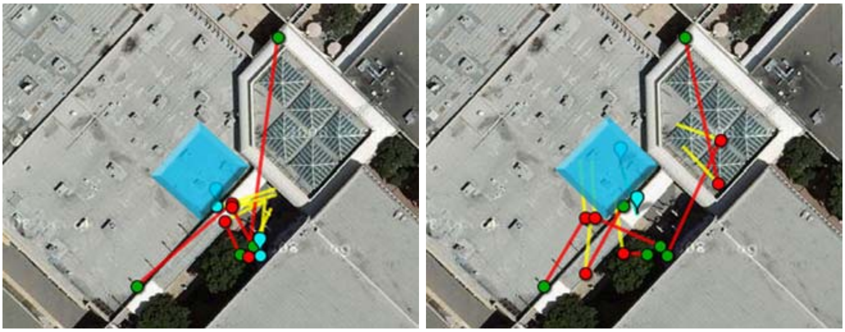


Fig. 10. Left image shows what happens if we apply the original algorithm to the landmark in the left image of Fig. 9. Large GPS error leads to bad pose estimation. Right image shows what happens if we do not perform outlier detection. We get proper orientation but scale is not estimated correctly. This demonstrates the need for our alternative alignment method.

If we were to apply the original alignment algorithm to the landmark shown in the left image of Figure 9, the GPS locations with a large error would be used to estimate the pose orientation (since they are far away from each other), and that would result in an incorrect orientation estimate. This is shown in the left image of Figure 10. The image on the right shows results for the same landmark if we do not eliminate outlier GPS locations. This leads to correct pose estimation, but incorrect scale estimation. This indicates the need for both corrections used in our alternative alignment method.

3 Future Work

Although we have made significant improvements over our previous system, we still have several areas of future work to explore.

Currently we have only used the detailed camera pose information for better augmentation of the images already present in the database or the live images taken by the user while navigating. However, this information could be further leveraged to produce zoomed-out and/or warped views showing more context and better viewing angle. Instead of rendering a directional arrow in the camera space of the image (which often do not contain a view of the desired path), we can render a view of the image using a virtual camera at the path location. This should provide more understandable views and can also include additional context. This approach is still lightweight enough for mobile devices since it uses few images, in contrast to full panoramas or complete 3D models. It also preserves the ability to highlight important landmarks along a route by not showing extraneous information. A conceptual view of this is shown in Figure 11.

Furthermore, the ability to register landmark geometry with the world could be used to increase visual fidelity of the navigation system. For example, the reconstructed geometry could be used to compute a landmark shape proxy. The database images of the landmark could then be projected on the shape proxy, resulting in a simple, compact and photo-realistic representation of the object akin to a surface light field [19]. This full 3D representation of the world would allow us to move away from giving navigation instructions as sequences of static

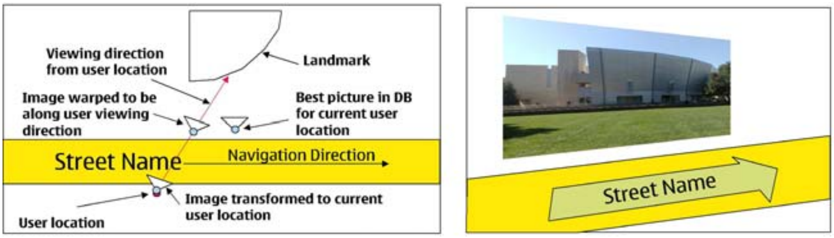

Viewing direction from user location
Landmark
Image warped to be along user viewing direction
Best picture in DB for current user location
Street Name
Navigation Direction
Image transformed to current user location
User location
Street Name

Fig. 11. This shows the concept of leveraging detailed camera pose information to produce zoomed-out view showing more context. The diagram on the left shows the camera location for the available image with respect to the desired camera position for the user. The right shows how this information is put together with additional context to create a representation for the new camera position.

images taken from fixed viewpoints to a continuous unconstrained 3D animation of the navigation route with total freedom in camera path selection.

We also currently take a complete navigation path as input, leaving the path planning to another system. We would like to include path planning in our system to produce a route between two points taking landmark information into account. Instead of simply generating the shortest route, we could generate the easiest to follow route or the most interesting route. Because individuals may have different preferences, we are also working on an adaptive framework to create a personalized user model for selecting better routes that contain more appropriate landmarks [20]. Finally, we plan to conduct user studies to evaluate the enhancements introduced in this work and those planned for the future in terms of qualitative improvements in user experience and clarity of navigation.

4 Conclusion

We have presented a pedestrian landmark-based navigation system with enhanced spatial reasoning. This work extends prior results in the area with new techniques to compute better choices of landmarks and more realistic augmentations of images, resulting in more natural navigation instructions. We enhance the system with a live-matching mode that augments the images of landmarks taken by the user while navigating.

Underlying these improvements are key computer vision technologies. The structure-and-motion reconstruction pipeline for computing the 3D landmark geometry and camera poses is necessary for better spatial reasoning and improved realism of image annotation. The robust image matching pipeline for mobile devices allows for a quick and reliable pose estimation of live images directly on the mobile device. A combination of these technologies leads to significant improvements in the quality of image selection, realism of their augmentation, and novel user-directed functionality. This work also shows a clear path for further improvements involving landmark-based navigation systems, some of which we discussed in the future work section.

References

1. Hile, H., Vedantham, R., Liu, A., Gelfand, N., Cuellar, G., Grzeszczuk, R., Borriello, G.: Landmark-Based Pedestrian Navigation from Collections of Geotagged Photos. In: Proceedings of ACM International Conference on Mobile and Ubiquitous Multimedia (MUM 2008). ACM Press, New York (2008)
2. Beeharee, A.K., Steed, A.: A Natural Wayfinding Exploiting Photos in Pedestrian Navigation Systems. In: MobileHCI 2006: Proc. of the 8th Conf. on Human-Computer Interaction with Mobile Devices and Services, pp. 81–88. ACM Press, New York (2006)
3. Millonig, A., Schechtner, K.: Developing Landmark-Based Pedestrian-Navigation Systems. IEEE Transactions on Intelligent Transportation Systems 8, 43–49 (2007)
4. Kolbe, T.H.: Augmented Videos and Panoramas for Pedestrian Navigation. In: Gartner, G. (ed.) Proceedings of the 2nd Symposium on Location Based Services and TeleCartography (2004)

5. Zhang, W., Košecka, J.: Image Based Localization in Urban Environments. In: International Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT 2006, North Carolina, Chapel Hill (2006)
6. Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y., Chen, W.C., Bismpigiannis, T., Grzeszczuk, R., Pulli, K., Girod, B.: Outdoors Augmented Reality on Mobile Phone using Loxel-Based Visual Feature Organization. In: ACM International Conference on Multimedia Information Retrieval (MIR 2008) (2008)
7. May, A.J., Ross, T., Bayer, S.H., Tarkiainen, M.J.: Pedestrian Navigation Aids: Information Requirements and Design Implications. Personal Ubiquitous Computing 7, 331–338 (2003)
8. Goodman, J., Gray, P., Khammampad, K.: Using Landmarks to Support Older People in Navigation. In: Brewster, S., Dunlop, M.D. (eds.) Mobile HCI 2004. LNCS, vol. 3160, pp. 38–48. Springer, Heidelberg (2004)
9. Kähäri, M., Murphy, D.: Sensor-fusion Based Augmented Reality with off the Shelf Mobile Phone. In: ISMAR 2008 Demo (2008), `http://ismar08.org`
10. Shankland, S.: Google Street View Goes Mobile. CNET Download (2008), `http://www.download.com`
11. Reitmayr, G., Schmalstieg, D.: Scalable Techniques for Collaborative Outdoor Augmented Reality. In: ISMAR 2004: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004) (2004)
12. Winter, S.: Route Adaptive Selection of Salient Features. In: Kuhn, W., Worboys, M.F., Timpf, S. (eds.) COSIT 2003. LNCS, vol. 2825, pp. 349–361. Springer, Heidelberg (2003)
13. Snavely, N., Seitz, S.M., Szeliski, R.: Photo Tourism: Exploring Photo Collections in 3D. In: SIGGRAPH Conference Proceedings, pp. 835–846. ACM Press, New York (2006)
14. Hile, H., Borriello, G.: Information Overlay for Camera Phones in Indoor Environments. In: Hightower, J., Schiele, B., Strang, T. (eds.) LoCA 2007. LNCS, vol. 4718, pp. 68–84. Springer, Heidelberg (2007)
15. Look, G., Kottahachchi, B., Laddaga, R., Shrobe, H.: A Location Representation for Generating Descriptive Walking Directions. In: IUI 2005: Proc of the 10th International Conference on Intelligent User Interfaces, pp. 122–129. ACM Press, New York (2005)
16. Lourakis, M., Argyros, A.: The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece (2004), `http://www.ics.forth.gr/~lourakis/sba`
17. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 91–110 (2004)
18. Szeliski, R.: Image Alignment and Stitching: A Tutorial 1. Technical Report MSR-TR-2004-92 (2005)
19. Chen, W.C., Bouguet, J.Y., Chu, M.H., Grzeszczuk, R.: Light Field Mapping: Efficient Representation and Hardware Rendering of Surface Light Fields. In: SIGGRAPH 2002: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, pp. 447–456. ACM, New York (2002)
20. Liu, A., Hile, H., Borriello, G., Kautz, H., Brown, P., Harniss, M., Johnson, K.: Informing the design of an automated wayfinding system for individuals with cognitive impairments. In: Third International Conference on Pervasive Computing Technologies for Healthcare, PervasiveHealth 2009 (to appear, 2009)