Location Based Services and TeleCartography, Series: Lecture Notes in Geoinformation and Cartography. G. Gartner, M.P. Peterson, and W. Cartwright (Eds.) Springer Verlag, Feb. 2007, pp. 393–416

„Anywhere Augmentation“: Towards Mobile Augmented Reality in Unprepared Environments

Tobias Höllerer, Jason Wither, and Stephen DiVerdi

Four Eyes Laboratory
Department of Computer Science
University of California
Santa Barbara, California, USA
holl@cs.ucsb.edu, jwither@cs.ucsb.edu, sdiverdi@cs.ucsb.edu

Summary

We introduce the term „Anywhere Augmentation” to refer to the idea of linking location-specific computing services with the physical world, making them readily and directly available in any situation and location. This chapter presents a novel approach to „Anywhere Augmentation“ based on efficient human input for wearable computing and augmented reality (AR). Current mobile and wearable computing technologies, as found in many industrial and governmental service applications, do not routinely integrate the services they provide with the physical world. Major limitations in the computer’s general scene understanding abilities and the infeasibility of instrumenting the whole globe with a unified sensing and computing environment prevent progress in this area. Alternative approaches must be considered.

We present a mobile augmented reality system for outdoor annotation of the real world. To reduce user burden, we use openly available aerial photographs in addition to the wearable system’s usual data sources (position, orientation, camera and user input). This allows the user to accurately annotate 3D features from a single position by aligning features in both their firstperson viewpoint and in the aerial view. At the same time, aerial photographs provide a rich set of features that can be automatically extracted to create best guesses of intended annotations with minimal user input. Thus, user interaction is often as simple as casting a ray from a firstperson

view, and then confirming the feature from the aerial view. We examine three types of aerial photograph features – corners, edges, and regions – that are suitable for a wide variety of useful mobile augmented reality applications. By using aerial photographs in combination with wearable augmented reality, we are able to achieve much higher accuracy 3D annotation positions from a single user location than was previously possible.

1 Introduction

A lost traveler walks down an unfamiliar street in a foreign country, looking for a specific address. His city map does not help him since all the street signs are in a foreign alphabet and he barely knows how to pronounce the street he is looking for. Fortunately, he carries an „Anywhere Augmentation“ device, which he uses as a pedestrian navigation tool to overlay position-specific directions immediately onto his field of view through his cell-phone display, which films and augments the scene in front of him. He interacts with the physical scene, pointing out cross streets, whose names, in response, appear directly on top of the physical world and are also highlighted in an optional map view of the neighborhood.

The leader of a reconnaissance team scouts out the terrain behind a hill that shields his troop from enemy view but also prevents them from surveying the landscape features and infrastructure behind it. He dons his augmentation glasses, and despite imperfect localization is quickly able to align the contours of the virtual 3D elevation model with the outline of the hill in front of him, establishing registration for overlaying the landscape and building features behind the hill directly in his field of view, in the fashion of Superman’s X-ray vision. He makes the decision on how to approach that terrain in a much more informed fashion.

A schoolchild is on a field trip to learn about botany. Distinguishing different orders and families of trees has never been his strength, but help is at his fingertips. His small electronic companion allows him to see labels of already classified trees directly overlaid on his view of them and allows him to add tentative new classifications and take pictures for later verification by the teacher or field guide.

What these three scenarios have in common is the idea of having computational resources available anytime and anywhere, and moreover, being able to form a link between the location-specific information and the physical world by direct vision overlays. We introduce the term „Anywhere Augmentation“ for this concept, emphasizing the necessity for

such a system to work in arbitrary environments in order to become adopted by users and make life easier for them. Currently, such technologies exist only for very limited example applications in research laboratories.

Mobile and wearable computing technologies have found their way into mainstream industrial and governmental service applications over the past decade. They are now commonplace in the shipping and hospitality industries, as well as in mobile law enforcement, to highlight but a few successful examples. However, current mobile computing solutions outside of research laboratories do not sense and adapt to the user's environment and they do not link the services they provide with the physical world. And because of major limitations in the computer's general sensing and scene understanding abilities and the infeasibility of instrumenting the whole globe with a unified sensing and computing environment, this is not likely to change soon, unless we find alternative approaches – which is the starting point for our work.

2 From Mobile Augmented Reality to „Anywhere Augmentation"

In spite of the great potential of mobile AR for many application areas, progress in the field has so far almost exclusively been demonstrated in a number of research prototypes. Actual commercial deployment is limited; early commercial technology and service providers are struggling to find customers and create markets. Despite better solutions to the technical challenges of wearable computing problem areas remain, such as the need for miniaturization of input/output technology and power supply, and for improved thermal dissipation, especially in small high-performance systems. Also, ruggedness is a key requirement. Outdoor AR is particularly challenging; in contrast to controlled environments indoors, one has little influence over outdoor conditions. Lighting can range from direct sunlight in a reflective environment to absolute darkness during the night. Outdoor systems should withstand all possible weather conditions, including wind, rain, frost, and heat.

However, it is not chiefly because of these issues that „Anywhere Augmentation" has not yet emerged as a widely adopted application technology. After all, many people operate their cell phones or their MP3 players comfortably, even under adverse conditions. It is already possible to manufacture hardware that can function in all the environments that we have in mind for „Anywhere Augmentation". First, however, it has to be

demonstrated that these devices can be usefully employed, and the problem is, at least to a certain extent, one of user interface design. With standard graphical user interfaces straightforwardly adapted from desktop and laptop computers, we have not seen sufficient user enthusiasm to warrant launching a whole new wave of wearable and situated computing computing products.

AR, which makes the physical world a part of the user interface experience, has the potential to play a significant role in changing this.

One of the main problems with current approaches to augmented reality is that in order to obtain reliable and accurate registration between the physical world and the augmentations one either needs a model of the environment, or the environment needs to be instrumented, at least passively with registration markers. Both of these preconditions severely constrain the applicability of AR. Instrumentation of environments on a global scale is exceedingly unlikely to take place, and detailed 3D city and landscape models are very cumbersome to create [1]. In addition, even if detailed 3D models of target environments existed on a broad scale, keeping them up-to-date would be a major challenge and they would still not take into account dynamic changes. Instead of relying on the existence of data that is not likely to become available in the near future, we propose to utilize several sources of GIS data for which there are already data repositories with nationwide coverage (e.g. aerial photography, elevation, land use, street maps, NGA names database). The general concept of „Anywhere Augmentation“ does not depend on the existence of any particular subset of these data sources, but instead we want to consider any of these sources when available, and their existence will improve the user experience by providing more information and stronger constraints for user interaction.

Annotation of outdoor scenes is an important part of mobile augmented reality research (cf. Fig. 1). Generally, the situated content displayed by a wearable system is carefully constructed offline using many different technologies, including modelling programs, GIS data, and aerial photographs. In this work, our focus is on annotating an outdoor scene from within the wearable system, providing an appropriate interface to allow accurate markup in a mobile context. To reduce the amount of manual work that must be done by the user, we have modified our system to use aerial photographs of the region in conjunction with the wearable's acquired data. This allows the user to accurately place 3D annotations from a single position by providing a means of accurately gauging depth.

Fig. 1. A wearable system for outdoor annotation. Left to right: (a) A user wearing the system to annotate an outdoor scene. (b) The user's first-person view, showing the scene from the ground and the visible annotations. (c) The user's overhead view, showing an aerial photograph of the local region with the user's position and placed annotations overlaid.

With orientation tracking, from a static position a user can easily cast a ray to select a visible feature in the scene, but setting the depth of that feature is more difficult. Previous work in this area requires the user annotate the same feature from multiple viewpoints to triangulate a position [22], or estimate depth from a static viewpoint using artificial depth cues [31]. However, commonly available aerial photographs [14, 32] can be used to allow accurate 3D position input from a single location. After a user has cast a ray, our system presents the user with an aerial view of the scene and the cast ray and allows the user to adjust the ray and set a distance. The result is a significant improvement in the accuracy of 3D positions over previous AR distance estimation work [31], as well as the ability to annotate features that may not be directly visible from the user's location, such as the opposite side of a building. Automatic feature extraction from the aerial photographs allows the system to intelligently recommend salient features along the cast ray, so the user needs only to choose from the detected features and possibly refine the result.

We examine three different types of features a user may want to place in the outdoor scene, based on how they appear in the aerial photograph. Corners can correspond to the vertices of building silhouettes and are useful for modelling geometry [4]. Vertical walls appear as edges that can be used to properly orient and position world-aligned billboard annotations [15]. Uniform regions in aerial photographs can denote buildings, fields, etc. and can be annotated with a label and a bounding box for wearable navigation purposes [12]. Our manual interface and automatic feature extraction techniques are thus geared towards finding these types of features in our aerial photographs. We use these annotations as a representative set of possible information a user may want to input, but our system is not geared towards any particular application and only minor modifications are needed to tailor the approach to other task scenarios.

A key focus of this work is the aggregation of available data sources, in this case the wearable's data streams and the aerial photographs, to reduce the burden on the user for traditional AR tasks. This is the first step towards our goal of anywhere augmentation, where the usual AR initial costs of manual modelling, calibration and registration are alleviated to make augmented reality readily available in any unprepared environment. Our contribution is to significantly reduce the work necessary to create physically-situated annotations in an unprepared, large-scale outdoor scene. The development and use of real-time, local, automatic feature extraction techniques for aerial photographs is a secondary contribution of this work.

3 Previous Work

The previous work for this project can be split between our two contributions. The first section compares our approach with other wearable systems dealing with outdoor annotation. The second section discusses feature extraction from aerial imagery.

3.1 Wearable Systems

Rekimoto et al. [25] introduced the idea of Augment-able Reality with a system that allows users to annotate the environment with contextual information at specific locations that had been prepared ahead of time with active or passive markers. They envisioned extending the system to allow annotations for any position of known GPS coordinates. Our system expands on this concept by allowing annotations of unprepared environments at arbitrary locations without known GPS coordinates.

More recent work has been done in using wearable systems to acquire accurate positions of arbitrary locations, towards the goal of modelling outdoor scenes from within a wearable system. Baillot et al.'s [4] wearable modeller is based on first creating 3D vertices by casting intersecting rays from multiple viewpoints, and then creating primitives fit to those vertices for the final model. Our annotations are a more easily acquired and more accurate version of their construction points, and could be used for the same sort of modelling application as they describe. The paper also shows an example using an indoor scene's architectural floor plan as a guide for creating vertices. This is the inspiration for our use of aerial photographs, but we extend the concept in many new directions – we use commonly available aerial photographs that are automatically registered to the user's

location, and we use them not only for creating points, but for many other types of annotations as well, and we can even use them to create accurate annotations for features that are not visible from the current location, such as placing a correctly oriented billboard label on the opposite side of a building.

Piekarski and Thomas' outdoor wearable modelling system [21, 22] implements a wide variety of techniques based around the concept of working planes that are created by sighting down the wall or feature to be modelled from one viewpoint, and then moving to another location to create points on that plane at the correct depth. Our solution replaces the need for working planes with the overhead view – working planes the user would normally have to construct from a particular location are already available as edge features in the aerial photograph, removing the need for the user to move around large buildings to distant locations for accurate modelling.

To avoid requiring multiple viewpoints for determining 3D positions, Reitmayr and Schmalstieg's system [24] has a complete model of the environment (obtained offline) that users can annotate by casting rays that intersect with the model's surfaces. The goal of „Anywhere Augmentation" aims to remove the initial start up costs associated with acquiring a detailed scene model offline – in our system, we remove the need for such a model by using aerial photographs to provide the same features for annotation.

Maps and aerial photographs have also been used in many mobile systems for passive localization purposes. The Cyberguide system [2] uses a hand-held device to display a rough estimate of the user's position and orientation on an abstract map with point-of-interest annotations. ARVino [17] uses aerial photographs in a virtual reality view of GIS data, to aid the user in mentally mapping abstract information onto the physical environment the data annotates. We extend the functionality of both Cyberguide and ARVino by using aerial photographs for passive localization of the user, as well as active annotation of the environment by the user and even for automatic extraction of new features.

Reitmayr et al. extract features from paper maps to display registered annotations with an overhead projector [23]. The type of features they use differ from ours in that they are geared towards robust identification and registration for displaying overlays, as opposed to our system which extracts user-specified semantic features as targets for annotation.

Finally, in place of a map, Bell et al. [7] and Güven and Feiner. [15] use a virtual model of the environment for localization by displaying a world in miniature view. This gives the user a more informed understanding of their surroundings, and could even be used to help users create situated

annotations. However, detailed offline model construction is not in the spirit of our goal of „Anywhere Augmentation“.

3.2 Feature Extraction

There is an extensive body of research in the realm of feature extraction from aerial photographs for scene understanding. Mayer [19] presented a thorough survey of automatic object extraction from aerial photographs, with an emphasis on buildings. This survey compares a wide range of projects in terms of their final output – while many of them create detailed building geometries, there are as many projects that focus on low complexity geometry and feature sets. A common theme of the more complicated papers surveyed is the use of sophisticated input data, such as two more images from multiple viewpoints [13], laser range data [3], or digital surface models [30]. Because of our emphasis on „Anywhere Augmentation“, we restrict our input to commonly available datasets, such as the single image, visible spectrum, near-orthographic aerial photographs from Google Maps [14] or Yahoo Maps [32].

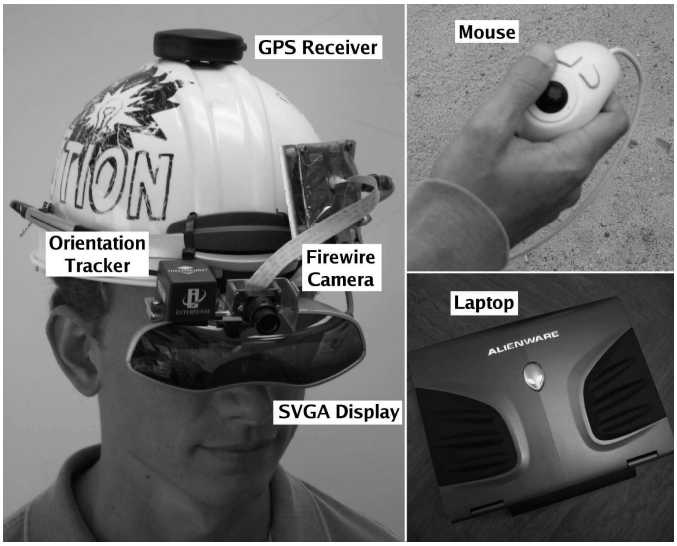

Fig. 2. Our wearable system hardware. An Alienware Area-51 m5500 laptop (worn in a backpack), an SVGA Sony Glasstron PLM-S700E display, a Point Grey Firefly firewire camera, An InterSense InertiaCube2 orientation tracker, a Garmin GPS 18 receiver, and an ErgoTouch RocketMouse.

The common trade off in automatic feature extraction is between lower complexity of input data and a higher computation cost. Many algorithms [28, 27, 18] rely on global analysis of region imagery and iterative minimization approaches that require significant compute time. Our goal of „Anywhere Augmentation“ discourages lengthy preprocessing on our datasets, which means that our feature extraction must be done on the fly, making costly algorithms impractical.

One observation that has proven useful in feature detection in a number of cases is that aerial photograph features are often correlated across different scales. Baumgartner et al. [6] use this property to search for different features of roads at different scales – lines at coarse resolutions, and uniform patches at fine resolutions – to generate an overall better model of road geometry. A slightly different approach was taken by Reitmayr et al. [23], by searching for the same features at multiple scales, taking advantage of the multiscale self-similarity of map data. Our corner extraction technique also utilizes multiscale detection based on the assumption that building silhouettes will result in salient corners at multiple scales, while noise and texture corners will not.

The performance of automatic building extraction algorithms, in terms of both speed and robustness, can be greatly improved by small amounts of user input. These semiautomatic approaches can require user input at the outset to guide detection, or after detection has occurred to correct errors. Vosselman and Knecht [29] and Kim et al. [16] both take the former approach towards the application of labelling roads for mapping. The user annotates a small section of road with the direction the road continues, which the algorithms are able to use to determine characteristics of the road and follow its path throughout the rest of the aerial photograph. Nevatia and Huertas [20] let the user correct errors in the results of their automatic building detection, and use the corrections to guide later detection steps. Our semiautomatic interface requires user input to determine the initial detection parameters, and then allows the user to manually correct errors in the detected results if necessary.

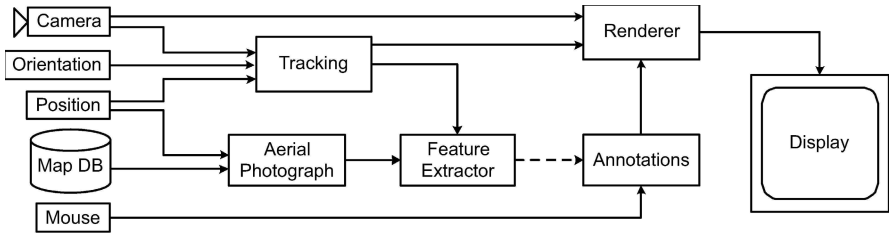

Fig. 3. An overview of our system structure. The camera, position, and orientation data are combined for a general tracking result. Position data is also used to query a database (Google Maps [14]) for the local aerial photograph. The aerial photograph and tracking data are used for feature extraction, which is optionally sent along with mouse input to the annotation model. Annotations, tracking, and the scene image are all used to render a final visual to the display.

4 System

Our wearable system can be seen in Fig. 2. At the core is an Alienware Area-51 m5500 laptop, which is worn on the user's back. The display is an SVGA Sony Glasstron PLM-S700E hanging from the front of the helmet, used in video see-through mode. Mounted directly above the display are a Point Grey Firefly firewire camera and an InterSense InertiaCube2 orientation tracker, and on top of the helmet is a Garmin GPS 18 position tracker. User input is through a hand-held ErgoTouch RocketMouse. All of these devices are relatively inexpensive, off-the-shelf components.

Many services now exist on the internet to provide access to aerial photographs and high-resolution satellite imagery of the world, including Yahoo Maps (at 1.0m resolution for the entire United States) [32] and Google Maps (with variable resolution) [14]. In our current system, we acquire 0.5m resolution aerial photographs offline from Google Maps and stitch them together into a single large view of the University of California, Santa Barbara campus. However, it is possible with a wireless internet connection to download map data on the fly based on the wearable's reported GPS coordinates. This automatic map acquisition would allow the system to work in new environments covered by map services without the initial setup cost.

See Fig. 3 for an overview diagram that shows all the components and connections of the complete system.

4.1 Calibration

While our position and orientation sensors report absolute coordinates in a global reference frame, both have too much error to be used without further calibration. We found that our GPS tracker can be off by as much as 10 meters, but that it also drifts slowly. To compensate for the initial offset users are asked to specify their exact location on the aerial photograph, which greatly reduces the position error for a single run of the application.

Our orientation sensor is designed to provide orientation information from true north, but has two major sources of error. First, the difference between true north and magnetic north is a systematic error, which we overcome by adding a second step to the calibration process. In this step the user centers the view at a distinct feature and then clicks a button to bring up the aerial photograph. The user's position and orientation are overlaid on the photograph, and the user is able to modify the displayed orientation to directly coincide with the user's chosen feature. This calibration procedure is similar to the single point calibration technique formalized by Baillot et al. [5], except that in our system the calibration location does not have to be predefined because the user chooses the necessary points on the aerial photograph during the process. However, because the user input is limited to adjusting the orientation in the overhead view, only error in yaw is accounted for. Roll and pitch can be roughly corrected by assuming the user's head is level and oriented vertically during the calibration.

The second source of error in the orientation tracker comes from nearby ferromagnetic materials. We found that this can distort our tracking results by as much as twenty degrees. To compensate for this, we integrated a modified version of our hybrid tracking system [31]. This system, which is based in part on previous work by Satoh et al. [26] and You et al. [33], corrects the orientation error by using gradient descent to find the best match between a set of points whose pose is updated by the inertial tracker and a matching set of points whose screen coordinates are tracked using a Lucas-Kanade optical flow feature tracker from OpenCV [10]. Individual image points are found via a corner finding algorithm from OpenCV, and the matching inertial points are computed by unprojecting the screen coordinates to a set distance from the user.

5 Annotation Interface

The intuition behind our use of aerial photographs to assist annotation is that they can fill the role of the second point of view necessary for triangulating 3D positions. Rather than having to walk to a different location and view the point again to find its depth, the user can instead find the point on an aerial photograph to provide the necessary information to calculate the distance to that point. This also provides a unified way of making many different types of annotations by marking up an aerial photograph – for example, specifying a corner, an edge or a region all correspond to well-understood 2D drawing tasks on an aerial photograph. Additionally, tasks such as specifying the perimeter of an entire building or labeling the rear wall of a building can be possible using aerial photographs, whereas they would otherwise require moving around to the building's opposite side. Our aerial photographs are always shown in a north-up orientation which has shown to be faster than a forward up configuration for search tasks like ours where the target is not shown on the aerial photograph [11].

While aerial photographs provide many opportunities for annotations by themselves, they are especially useful in combination with a wearable system. With only an aerial photograph, it is possible to annotate many types of features, but only in 2D – modelling the accurate height of a building would be very difficult. Our wearable system provides an advantage by allowing the specification of a 3D position from the combination of the aerial photograph with the first person viewpoint. The usefulness of aerial photographs is also greatly increased when the user can be situated in the environment they are annotating. For example, it may be difficult to distinguish features in an aerial photograph alone, but when a user can stand in the scene and look at the buildings from a ground-level viewpoint, these ambiguities can be more easily resolved. Having both the aerial photograph and the first person view is analogous to having a perspective view and a top-down view in a CAD modeller – while many things can be done in either view independently, having both views is often faster and more powerful.

Our annotation system utilizes these two views of the scene to make specifying annotations very easy. First, the user casts a ray in the direction of the feature to annotate by centering it in the field of view in first person mode. Once a ray has been cast, the view switches to an aerial photograph mode, with the user's position and the ray overlaid. Then only a few simple interactions are necessary for the user to create any type of annotation at the correct location in the overhead view.

The interaction techniques we chose to use for annotating are focused on the ray that is cast from the first person view. To place an annotation, the user casts a ray in the appropriate direction, and then sets the distance of the annotation along that ray. We chose to break this interaction into two one-dimensional tasks to reduce user burden and increase the resulting precision in the less-precise wearable environment [8], and to keep the ray cast by the user central to the annotation interaction. Instead of using mouse input, it would also be possible to interact with the aerial photograph using a tablet PC. However, it has been shown that while stylus interactions are faster in a wearable environment, they are less precise as well [9].

The three types of annotations we examine are corners, edges and regions. See Fig. 4 for examples of each of these features in the aerial photograph view. The specifics of the interface for each type of annotation, as well as example applications for each type of annotation are described in the three sections below.

5.1 Corners

The most general type of feature, corners can be used for many different applications. They are not limited to corners of buildings but can represent any feature that has a distinct, visible location in the aerial photograph. This could include objects like light poles along a street, doors of buildings (if paths lead to them), trees, or even features on the ground like street or sidewalk corners.

The most straightforward application for corner annotations is modelling. These corners could be used like Baillot et al.'s construction points [4], or as a sparse point cloud in any other modelling application. Corners could also simply be used as 3D points to bind contextual information to.

Fig. 4. The aerial photograph view from within the system. In the lower left corner, an insert of the video feed from the head-worn camera can be seen. The user's position and orientation are represented on the photograph with a small cone avatar. A small set of features have already been annotated – two corners (the green points), one edge (the green line), and one region (the transparent green rectangle).

Placing a corner in 3-space is a three step process in our system. First the user finds the feature they want to annotate and centers that feature in their field of view. A mouse click changes to the aerial photograph view, with the user's position and the cast ray drawn on top. If tracking error has caused the ray not to intersect the selected feature, the user can change its direction by rolling the trackball in the direction the user wants the ray to move. When satisfied, another mouse click creates an annotation on the ray at the user's position. Rolling the trackball away from the user moves the annotation further away along the ray – once the annotation is at the same distance as the feature, a final mouse click completes the annotation. The view returns to first person mode and the user can see their corner annotation as a small cube (see Fig. 5a). An important note here is that the annotation will appear at the correct height in the first person view, because the user originally cast a 3D ray in the first step.

5.2 Edges

Edges in aerial photographs are useful for many different kinds of annotations. Multiple edges could be used to model anything with sharp image boundaries, such as building perimeters, fields, pools, sidewalks and roads. Another use for annotating edges is placing world registered billboard labels. For instance Güven and Feiner [15] use world-stabilized images in their authoring environment to localize the information they are presenting, by displaying the annotation as a billboard on an existing structures.

Fig. 5. Example annotations as seen by the user in first person view mode. Left to right: (a) Corner annotations on the corners of two buildings are rendered as cubes. (b) An edge is annotated with a texture mapped onto the plane of the wall it denotes. (c) A region annotation is rendered as a wireframe bounding box. These renderings are not geared towards a particular application; rather, they are for illustrative purposes. Applications using these annotations would have visual representations tailored to their needs.

Creating an edge annotation in our system follows the same basic procedure as creating a corner annotation. The user centers the edge to be annotated in first person view, adjusts the cast ray and sets the distance along the ray. Instead of a point, the edge annotation is drawn as a line segment perpendicular to the cast ray. Once the annotation is positioned correctly, a final step is needed to adjust its orientation to align with the feature being annotated. This is done in the same way the cast ray is adjusted, by moving the trackball in the desired direction of rotation. After the annotation is fully specified, the display returns to the first person view where the user can see the new annotation (see Fig. 5b).

5.3 Regions

Regions are the third type of feature we have chosen to annotate. An example of why region annotations are useful is given by Feiner et al. [12], who use screen oriented, world stabilized annotations to label buildings.

These annotations can be particularly useful if tracking is not robust enough to support more tightly registered annotations such as edge or corner annotations. We also give our region annotations a width and depth (the x- and y-axis on the aerial photograph, respectively), so regions can very quickly be used as a rough axis-aligned model. This is obviously most useful when the buildings are also rectangular and axis-aligned. Then, the user can get a simple but complete model by casting a ray towards the roof of a building to give the annotation the appropriate height as well as width and depth. Generally, any large aerial photograph feature could be usefully annotated by regions, such as fields and parking lots, or even more visually complex semantic regions like a park full of trees or a group of buildings.

Specifying a region annotation follows the same basic steps as the corner annotation. This time, the user casts a ray through the center of the area to annotate and sets the distance along the ray so the final position is at the center. To finish the region, the user then drags out a corner of the bounding box to fit the area on the aerial photograph, and that action is mirrored for the other three corners of the bounding box. The result is the region annotation bounds the area in the aerial photograph, and its height is set to the height of the original ray at the distance to the annotation's center. Afterwards, the display is returned to the first person view, where the region annotation is drawn as a wireframe box (see Fig. 5c).

Fig. 6. The region of the aerial photograph searched for features. The user's view is represented by a small cone avatar. The white rectangle is the local portion of the aerial photograph the filters are applied to, and the dotted red lines show the region searched for for valid features. The angle swept by the dotted red lines is equal to twice the expected orientation error.

6 Feature Extraction

In addition to providing a useful viewpoint for users to manually annotate an outdoor scene, aerial photographs also provide a great deal of information that can be automatically segmented with appropriate image processing. We leverage this information by attempting to automatically detect the feature the user is annotating. If the feature is detected correctly, the user only needs to confirm it in the overhead view – otherwise, the same selection interface can be used to correct any errors in the detected feature. Thus, the semiautomatic approach does not significantly add complexity to the interface, but does frequently reduce the amount of input necessary, significantly reducing the burden on the user.

The main limitation of our automatic feature detection is that it must be fast, as it is executed each time the user casts a ray. To reduce the amount of work required each step, the detection algorithm is only run on a small search region that contains the ray cast from the user's position out to a maximum distance, rather than the entire aerial photograph (see Fig. 6). This reduces the amount of data to be processed at any given time by an order of magnitude. We also restrict the feature detectors to use only fast, local filters, instead of global or pairwise-pixel operators. Given these constraints, performance of the filters has not been a problem for the user experience.

Small errors in the tracking and the imperfectness the feature detection necessitate a certain amount of flexibility in the set of detected features returned. To address orientation tracker error, an angular epsilon term is used to define a search cone around the cast ray in which valid features may be found (see Fig. 6). For the case that the best detected feature is not the intended feature, the best n features are returned.

The annotation interface must be slightly modified to support automatic feature detection. After the user centers a feature in the field of view and casts a ray, in the aerial view of the scene, multiple detected features along the ray are presented. The user can then easily scroll through the options in the order of distance from the user by rolling the trackball up or down to select one. If the selected annotation is accurate enough, the user confirms it by clicking the middle mouse button and the view returns to first person mode. Otherwise, the user clicks the right mouse button and is presented with the same interface to adjust the annotation as for manual positioning.

6.1 Corners

Because of the relatively large size of buildings with respect to other features in an outdoor environment, building corners have the convenient property that they are persistent at a large number of different scales of the same image, whereas noise corners that come from texture, image noise, or small objects, disappear at coarser scales. On the other hand, at coarser scales, some pixels may combine to create new false corners, and small, distinct objects with valid corners may be lost entirely. Therefore, a multiscale approach to corner detection will provide more robust results for all sizes of corners a user may want to annotate. Our basic approach is to generate a corner image of the local region the user is annotating, where each pixel represents the likelihood that that pixel is a corner, by using OpenCV's Harris corner detector the region at multiple scales on multiple scales and summing the results (see Fig. 7a). Local maxima of the smooth corner function are extracted by a sliding 5x5 window. Then, the region along the user's cast ray is searched for the maximum weight pixels (from the set of local maxima). The weighting function is

$$w = w_s * w_a * w_d \quad (1)$$

where ws is the strength of the corner sampled from the corner image, wa is an angular term and wd is a distance term. wa is computed by finding the angular offset to a pixel from the cast ray and interpolating the weight from one to zero as the offset goes from zero to the angular epsilon. wd is determined by the distance to the pixel – it is set to one past a minimum distance threshold, and interpolated between zero and one within the threshold. An example set of detected corner features can be seen in Fig. 7b.

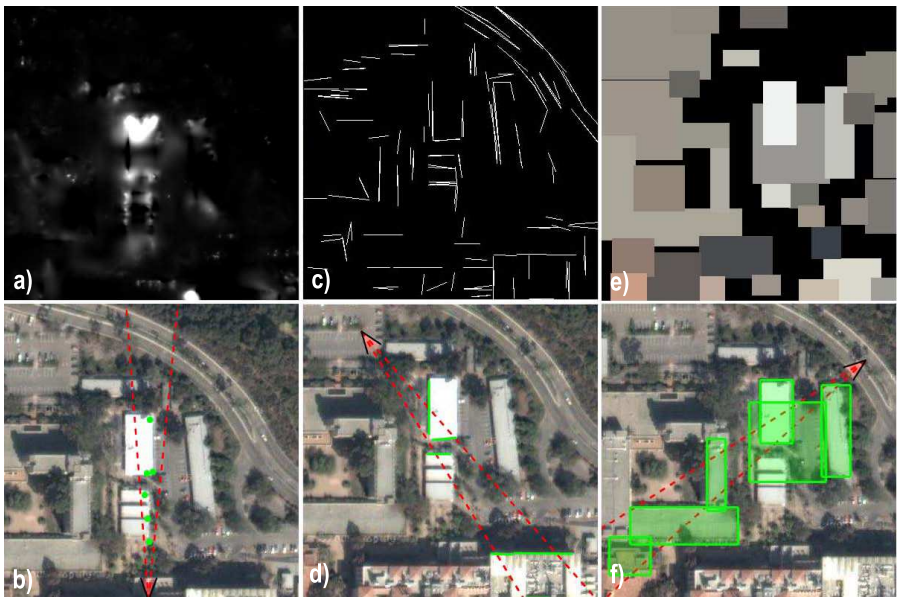

Fig. 7. Outputs of each of the automatic feature detectors. Top to bottom, left to right, : (a) The Harris corner transform is applied at multiple scales and summed together. (b) Corner features are selected from the local maxima of the continuous function. (c) Coherent line segments extracted from the output of Canny edge detection. (d) Edge features are selected from these line segments, weighted by their gradient magnitude. (e) After segmenting uniform color regions, components are merged and represented by their bounding boxes. (f) Region features are selected from components by size, intersection and distance from the center.

6.2 Edges

Finding edges in the search region is a simple matter of using OpenCV's Canny edge detection operator. However, obtaining the most salient edges from that image is more complicated. Our approach is to find the set of connected contours in the edge image and use OpenCV's polygon approximation algorithm to simplify the contours to within a certain error threshold. After the simplified contours are generated, all segments below a minimum length threshold are removed. The minimum length is linearly interpolated between a threshold for dominant-direction edges (0 or 90 degrees), and a different threshold for diagonal edges (45 degrees off the dominant directions). This is based on the observation that buildings tend to have 90 degree corners and tend to be oriented in the same direction as one another – for our example dataset, most of the buildings are axis-aligned – whereas noise edges tend to be randomly oriented. If there is no

correlation among building orientations, the two thresholds can be set to the same value, removing the bias.

Once the final set of edges is determined (see Fig. 7c), the weight of each edge is calculated and the maximum weight edges are returned. The weighting function is

$$w = w_i * w_o * w_d * w_g \quad (2)$$

where wi is an intersection term, set to zero if the edge does not penetrate the cast ray's error region, interpolated up to one if the edge intersects the actual cast ray. wo is an orientation term, interpolated between one if the edge is perpendicular to the cast ray, and zero if the edge is parallel. wd is the same distance term as used for the corner detection – if the edge is within a minimum distance threshold, it is weighted lower. The last term, wg is a strength of the edge, determined by sampling the magnitude of the image gradient at the edge's midpoint. The gradient is computed using OpenCV's Sobel filter along the x and y dimensions. An example of the detected edges in a region can be seen in Fig. 7d.

6.3 Regions

As region features are areas like building rooftops, fields, parking lots, etc., they often appear in aerial photographs as areas of uniform pixel values plus some local texture, surrounded by a boundary edge. To find regions then, the first step is to attempt to reduce the influence of local texture by using OpenCV's closure morphology operator. The Canny edge detector is then applied to the texture-suppressed image to find region borders. Ensuring connectedness of the boundaries is necessary, so the closure operator is applied to the edge image, and then thinning is done to restore single-pixel wide edges. The result is a binary image that segments the aerial photograph into regions of similar color. The components are extracted using a flood fill algorithm, and basic component metadata is computed including average color and bounding box. This tends to produce many small components, with buildings and fields split up across multiple regions. To combine these small components we first cull some components based on HSV representation – since we focus on buildings, grass and tree areas are distractions, so any components with a basic green appearance are thrown out (this application-specific simplification could easily be replaced by something more sophisticated such as a clustering approach to vegetation segmentation). Then components are combined

based on the percentage overlap of their bounding boxes and their perceptual color similarity as calculated by the euclidean distance in CIE L*a*b* color space, and components that are below a minimum area threshold are discarded. The final list of components (see Fig. 7e) is then weighted and the maximum weight components are returned. The weight function is

$$w = w_i * w_a * w_p * w_d \quad (3)$$

where wi is a binary intersection term – if the ray is cast from within the region, or if the ray does not intersect the region, it is set to zero, otherwise one. wa is an area weight term, interpolated from zero to one between a minimum and maximum area value. wp is the perpendicular term, calculated as the perpendicular distance between the center of the region and the cast ray, as a percentage of the region's diagonal length. wd is the same distance weight term as the corner and edge detectors, penalizing regions that are too close to the user. A typical set of detected region features can be seen in Fig. 7f.

7 Discussion

Informal testing shows that the use of aerial photographs allows users to annotate scene features in 3D from a static viewpoint with much greater precision than was previously possible [31]. The longitude and latitude accuracy of an annotation position is limited only by the accuracy of the map and the ability of the user to manipulate the position accurately. Google Maps provides data at 0.5m per pixel resolution for Santa Barbara [14], and user input is generally accurate within a few pixels, so our final annotation precision is §1.5m. While additional interface modifications may make pixel-accurate user input possible, it is unlikely precision would increase as expected since photograph data often has noise and blurring that make sharp features like corners and edges appear to occupy multiple pixels. For some feature types, automatic, local energy minimization could potentially provide subpixel accuracy. Since height information is computed from the ray cast by the user, its accuracy is dependent on the quality of the orientation tracking.

The performance of the automatic feature extraction was informally tested in an offline environment. For each type of annotation, 7 user positions were selected from a large area aerial photograph, and for each location, multiple visible features were targeted to annotate (57 corners, 34

edges and 31 regions). The detected features were inspected, and if any were close enough to the intended feature that manual correction would not be necessary, it was recorded as a success. The results of these tests were that corner detection was successful approximately 65% of the time, as was edge detection, while region detection had a success rate of approximately 40%. We want to make very clear that even a 40% success rate leads to a substantial speedup of user interaction in the general case since there is no considerable penalty to pay for "failed" feature preselection. In the worst case, the user simply resorts to completely manual selection.

Once annotations are placed, the accuracy of their appearance in the first-person augmented reality view is determined by our system's tracking accuracy. Even with our hybrid vision and gyroscopic orientation tracking, there is still error up to five degrees from the automatic acquisition and reintroduction of new vision features. Standard PC GPS units make cheap, wide-area position tracking possible, but at low accuracy. Our system regularly experiences drift of up to a few meters over short periods of time even in clear conditions. Differential GPS or a hybrid GPS and vision or inertial position tracker would improve this result. These small position and orientation errors result in apparent mismatch between annotations and image features in first-person view, even with good annotation position accuracy.

Aerial photographs bring with them a number of limitations. First of all, while we use nearly top-down orthographic images, there is still a slight off-axis view angle that causes the roofs of tall buildings to shift a small, non-uniform distance (up to five pixels) from their ground perimeter. Currently, we do not account for this effect. Actual top-down orthographic aerial photographs or more sophisticated image processing would alleviate this problem.

The resolution of aerial photographs limits what sort of features they are useful for annotating. Small objects such as lamp posts, flag poles, or picnic tables may occupy too few pixels to extract multiple corners, or may even not appear recognizably at all. Higher resolution aerial imagery is steadily becoming more available and will help address this problem. However, aerial photographs are also captured infrequently, sometimes with many years elapsing between updating regions. This means that non-architectural objects such as picnic tables, cars, bike racks, temporary installations will not be represented or will be represented inaccurately. More troublesome can be new buildings that do not appear at all in a photograph.

8 Conclusion

We present a novel mobile augmented reality system for the annotation of features in large, outdoor scenes. Our primary contribution is the integration of a new data source, aerial photographs, to significantly reduce user burden while increasing annotation accuracy. Our secondary contribution is the use of real-time, heuristic automatic feature extraction on aerial photographs to further reduce user burden. The end result is a significantly improved interface and user experience for the traditional augmented reality task of outdoor annotation.

There are many avenues for improvement in future work. Foremost is the opportunity to include additional data sources, such as elevation data, road maps, and other GIS data. Fusion of many different, commonly available sources will improve the robustness and general applicability of our technique. Examining a larger array of annotation types would allow for a more complete model of the scene with more general usefulness in a wider variety of applications. A greater level of sophistication in the automatic feature extraction would further reduce the user burden and could improve the overall time required to model a scene. Finally, better tracking technologies to stabilize GPS drift and reduce orientation inaccuracy would enhance the user experience in first-person mode after annotations have been placed. We are currently developing a cheap tracking modality with no significant setup requirements based on a small camera pointing towards the ground so as to analyze optical flow in the way an optical mouse does on a smaller scale. By itself, this yields high frequency, high resolution relative position information similar to an inertial navigation system, but with significantly less drift. When coupled with a wide area tracking modality via a complementary kalman filter, the hybrid tracker becomes a powerful base for indoor and outdoor mobile mixed reality work.

Most important for us among all these future work opportunities is to remain true to the goal of „Anywhere Augmentation“. The work must continue to emphasize low startup cost and quick, easy integration to new scenes. This way, the traditional barriers to high quality augmented reality can be overcome, significantly increasing the general appeal of augmented reality solutions.

Acknowledgments

Special thanks to Ingrid Skei and John Roberts for their work on the initial prototype. This research was in part funded by a grant from NSF IGERT in Interactive Digital Multimedia #DGE-0221713, and an equipment donation from Microsoft.

References

1. Mahdi Abdelguerfi. 3D Synthetic Environment Reconstruction. Kluwer Academic Publishers, New York, NY, June 2001.
2. G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: A mobile context-aware tour guide. In Proceedings of Wireless Networks, pages 421–433, 1997.
3. P. Axelsson. Integrated sensors for improved 3D interpretation. International Archives of Photogrammetry and Remote Sensing, 32(4):27–34, 1998.
4. Y. Baillot, D. Brown, and S. Julier. Authoring of physical models using mobile computers. In Proceedings of the International Symposium on Wearable Computers, pages 39–46, 2001.
5. Y. Baillot, S. Julier, D. Brown, and M. Livingston. A tracker alignment framework for augmented reality. In Proceedings of the the International Symposium on Mixed and Augmented Reality, pages 142–150, 2003.
6. A. Baumgartner, C. Steger, H. Mayer, and W. Eckstein. Multi-resolution, semantic objects, and context for road extraction. In Proceedings of Semantic Modelling for the Acquisition of Topographic Information from Images and Maps, pages 140–156, 1997.
7. B. Bell, T. Höllerer, and S. Feiner. An annotated situation-awareness aid for augmented reality. In Proceedings of User Interface Software and Technology, pages 213–216, 2002.
8. D. Bowman. Interaction Techniques for Common Tasks in Immersive Virtual Environments. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 1999.
9. A. Chamberlain and R. Kalawsky. A comparative investigation into two pointing systems for use with wearable computers while mobile. In Proceedings of the International Symposium on Wearable Computers, pages 110–117, 2004.
10. Intel Corporation. Open Source Computer Vision Library Reference Manual. Intel Corporation, 2000.
11. R. Darken and H. Cevik. Map usage in virtual environments: Orientation issues. In Proceedings of Virtual Reality, pages 133–140, 1999.
12. S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban

environment. In Proceedings of the International Symposium on Wearable Computers, pages 74–81, 1997.
13. A. Fischer, T. Kolbe, F. Lang, A. Cremers, W. Förstner, L. Plümer, and V. Steinhage. Extracting buildings from aerial images using hierarchical aggregation in 2D and 3D. Computer Vision and Image Understanding, 72(2):185–203, 1998.
14. Google maps, 2006. http://maps.google.com/.
15. S. Güven and S. Feiner. A hypermedia authoring tool for augmented and virtual reality. In Proceedings of the International Symposium on Wearable Computers, pages 89–97, 2003.
16. T. Kim, S. Park, T. Kim, S. Jeong, and K. Kim. Semi automatic tracking of road centerlines from high resolution remote sensing data. In Proceedings of Asian Conference on Remote Sensing, 2002.
17. G. King,W. Piekarski, and B. Thomas. ARVino – outdoor augmented reality visualisation of viticulture GIS data. In Proceedings of the International Symposium on Mixed and Augmented Reality, pages 52–55, 2005.
18. C. Lin and R. Nevatia. Building detection and description from a single intensity image. Computer Vision and Image Understanding, 72(2):101–121, 1998.
19. H. Mayer. Automatic object extraction from aerial imagery – a survey focusing on buildings. Computer Vision and Image Understanding, 74(2):138–149, 1999.
20. R. Nevatia and A. Huertas. Knowledge-based building detection and description. In Proceedings of the Image Understanding Workshop, pages 469–478, 1998.
21. W. Piekarski and B. Thomas. Interactive augmented reality techniques for construction at a distance of 3D geometry. In Proceedings of the Workshop on Virtual Environments, pages 19–28, 2003.
22. W. Piekarski and B. Thomas. Augmented reality working planes: A foundation for action and construction at a distance. In Proceedings of the International Symposium on Mixed and Augmented Reality, pages 162–171, 2004.
23. G. Reitmayr, E. Eade, and T. Drummond. Localisation and interaction for augmented maps. In Proceedings of the International Symposium on Mixed and Augmented Reality, pages 120–129, 2005.
24. G. Reitmayr and D. Schmalstieg. Collaborative augmented reality for outdoor navigation and information browsing. In Proceedings of the Symposium on Location Based Services and TeleCartography, pages 31–41, 2004.
25. J. Rekimoto, Y. Ayatsuka, and K. Hayashi. Augment-able reality: Situated communication through physical and digital spaces. In Proceedings of the International Symposium on Wearable Computers, pages 68–75, 1998.
26. K. Satoh, M. Anabuki, H. Yamamoto, and H. Tamura. A hybrid registration method for outdoor augmented reality. In Proceedings of the International Symposium on Augmented Reality, pages 67–76, 2001.

27. J. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. Transactions on Pattern Analysis and Machine Intelligence, 21(3):282–288, 1999.
28. G. Sohn and I. Dowman. Extraction of buildings from high resolution satellite data. Automated Extraction of Man-Made Objects from Aerial and Space Images, 3:339–348, 2002.
29. G. Vosselman and J. Knecht. Road tracing by profile matching and kalman filtering. In Proceedings of Ascona Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images, pages 255–264, 1995.
30. U. Weidner. Digital surface models for building extraction. Proceedings of Automatic Extraction of Man-Made Objects from Aerial and Space Images, 2:193–202, 1997.
31. J.Wither and T. Höllerer. Pictorial depth cues for outdoor augmented reality. In Proceedings of the International Symposium on Wearable Computers, pages 92–99, 2005.
32. Yahoo maps, 2006. http://maps.yahoo.com/.
33. S. You, U. Neumann, and R. Azuma. Hybrid inertial and vision tracking for augmented reality registration. In Proceedings of Virtual Reality, pages 260–268, 1999.