2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Orientation Descriptors for Localization in Urban Environments

Philip David and Sean Ho
Army Research Laboratory
Adelphi, MD 20783
{phild,sean}@arl.army.mil

Abstract— **Accurately determining the position and orientation of an observer (a vehicle or a human) in outdoor urban environments is an important and challenging problem. The standard approach is to use the Global Positioning System (GPS), but this system performs poorly near tall buildings where line of sight to a sufficient number of satellites cannot be obtained. Most previous vision-based approaches for localization register ground imagery to a previously generated ground-level model of the environment. Generating such a model can be difficult and time consuming, and is impractical in some environments. Instead, we propose to perform localization by registering a single omnidirectional ground image to a 2D urban terrain model that is easily generated from aerial imagery. We introduce a novel image descriptor that encodes the position and orientation of a camera relative to buildings in the environment. The descriptor is efficiently generated from edges and vanishing points in an omnidirectional image and is registered to descriptors previously generated for the terrain model. Rather than constructing a local CAD-like model of the environment, which is difficult in cluttered environments, our descriptor measures, at equally spaced intervals over the 360° field of view, the orientation of visible building facades projected onto the ground plane (i.e., the building footprints). We evaluate our approach on an urban data set with significant clutter and demonstrate an accuracy of about 1 m, which is an order of magnitude better than commercial GPS operating in open environments.**

I. Introduction

Localization is a key component in many augmented reality, smart phone, and mobile robot applications. The standard approach for outdoor localization is the Global Positioning System (GPS). Commercial GPS provides accuracy on the order of 10 m, and when corrections are available, it can be as good a 1 cm. However, GPS requires an open environment so that the receiver may obtain a line of sight to a sufficient number of satellites. This is often not possible in dense, urban environments. Global localization in GPS-denied environments is an open problem and a very active area of research. Alternate approaches employ a variety of sensors, including dead reckoning, inertial, stereo, LIDAR, cellular radio, and visual. As no sensor or algorithm provides perfect localization in all environments, a combination of sensors and algorithms will be needed. GPS-denied localization is challenging due to the complex, cluttered nature of many environments, the changes in illumination that take place at different times, and the lack of high-fidelity 3D urban terrain models. As cameras are small, cheap, and are becoming ubiquitous, our focus in this paper is the development of a new model-based visual localization algorithm for outdoor urban environments.

Vision-based global localization can be accomplished by recognizing landmarks (i.e., immobile objects: buildings, bridges, statues, etc.) and retrieving their prerecorded positions from an existing database or terrain model. Appearance-based approaches represent the environment as collections of 2D images. Model-based approaches represent the environment in terms of higher level structures (*e.g.*, 2D or 3D geometric models). A substantial effort is required to create the image databases for appearance-based landmark recognition in large urban environments: the entire environment must be traversed while omnidirectional geo-tagged imagery is recorded. This may be impractical due to cost, traffic restrictions, or a requirement to quickly deploy a localization system. For these reasons, we are interested in using high-altitude aerial imagery to model urban environments. Aerial images have the advantages that they are readily available for most of the world and a single aerial image can be used to quickly generate a 2D map for a very large region of a city. We believe the ability to accurately localize and navigate an urban environment using 2D maps will provide a significant increase in capability of future location-aware mobile applications.

Given a high resolution aerial image of an urban environment and a omnidirectional query image taken on the ground level, our goal is to determine the position on the ground plane, and the orientation, of the camera. Our terrain model is a 2D map generated from an aerial image that consists of the footprints of all buildings in the area of interest; a building footprint is the projection of a building's vertical facades onto a horizontal ground plane. Using the 2D map, we compute, in an off-line process, building footprint orientation (FPO) descriptors over a uniform 2D grid of points on the map. In the on-line localization process, given an omnidirectional image for an unknown query view, the vanishing points in the image are used to detect and project the visible vertical building facades onto the ground plane and from these projections the query image's FPO descriptor is computed. The query FPO descriptor is compared to the terrain model FPO descriptors in a search for the position and orientation that minimizes their distance. The FPO descriptor and distance function are designed to be tolerant to a high degree of scene clutter and occlusion.

Our main contribution is the introduction of the FPO descriptor that provides a 2D geometric description of a local environment as observed from a single point in the

scene. Unlike previous approaches that use aerial images for localization, we do not require an explicit reconstruction of the structure of the local environment, which is often difficult in complex environments. The FPO descriptor, which is tolerant to significant scene clutter, combined with our robust distance function, results in accurate pose estimation in complex urban environments. We evaluate our approach in a challenging urban environment and show improved accuracy compared to existing 2D model-based approaches.

II. Related Work

Approaches to localization can be roughly categorized as either global or incremental. Global localization methods assume that the camera's pose is initially unknown while incremental methods assume the initial pose is approximately known and then refine that pose given new sensor data. There is overlap between the two approaches as many global localization algorithms are only effective when the pose of the camera can be confined to a relatively small part of the environment. Global localization methods may be further divided into metric and topological methods. Topological localization (*e.g*., [1], [2], [3], [4]) provides place recognition rather than metric location in a map; topological localization may be used by metric localization approaches to confine the region of a map that needs to be searched. Simultaneous Localization and Mapping (SLAM, [5], [6]), using monocular, stereo, or range cameras, are incremental localization approaches that build 2D or 3D models of the environment as a sensor moves and simultaneously determine the sensor's pose in the environment. In general, incremental algorithms run continuously as a camera moves in order to determine a camera's pose at any later time. Conversely, global approaches only require camera images to be processed when and where the camera's pose is needed. As our method is a global approach, we discuss only related global metric localization approaches in the remainder of this section .

Appearance based techniques (*e.g*., [7], [8], [9], [3], [10], [11], [12]) represent an environment as a database of geo-tagged key images and estimate the pose of a camera using the key images that best match the query image. Many of these techniques begin with a topological localization step. Zhang and Kosecka [13], [9] use wide baseline matching of SIFT ([14]) features to identify a few close key images in the database and then use triangulation to estimate the metric location of the camera. Se *et al*. [7] use SIFT features for both appearance-based global localization and for incremental 3D SLAM. Johns and Dudek [8] match images using 2D skyline contours. Using geo-tagged images on the world wide web (*e.g*., [12]) has recently become popular. Yeh *et al*. [2] combine image and textual searches of the web to perform localization. Although a wide variety of appearance based methods have been developed, they have difficulty when many different buildings have similar appearance due to a common architectural style, or when there are large changes in illumination between the model and query images. Also, appearance-based image databases are generated by traversing an entire urban environment in advance of deploying the localization systems; this is impractical when the localization systems must be hastily deployed but an image database doesn't already exist.

Model-based techniques represent an environment with a 2D or 3D model and then register the query images to the model. Some authors (*e.g*., [15], [16], [17]) estimate 3D camera orientation using vanishing points and the assumption that most edges in the environment are aligned with three orthogonal directions corresponding to a local world coordinate system. Ramalingam *et al*. [18] match skylines extracted from omnidirectional images to skylines generated from 3D wireframe city models. Lee *et al*. [19] estimate camera pose using vanishing points to constrain the matches of 2D image lines to lines in 3D wireframe building models. While 3D models exist for some large urban environments, most cities lack such models, so use of 2D models is more desirable.

A number of existing approaches perform localization by registering ground imagery to 2D urban models. 2D models are relatively easy to generate from aerial imagery, either manually or automatically, and are therefore more attractive than 3D models for urban environments. McHenry *et al*. [20] use visual odometry and a particle filter to match 3D rooflines generated from a stereo camera to a 2D terrain model generated from aerial imagery. Leung *et al*. [21] use vanishing points to infer the 3D orientation of building boundaries and apply a particle filter to estimate pose; they require pose filtering of a moving camera in order to obtain accuracy comparable to commercial GPS. The work most similar to our approach is that of Cham *et al*. [22] who analyze the vanishing points in omnidirectional images to create 3D local structural models from vertical building corners and neighboring plane normals, and then match these to a 2D urban model. Cham *et al*. report that their accuracy was limited due to difficulties in creating a sufficient number of accurate 3D local structural models. Our approach is similar to these in that it makes use of 2D terrain models generated from aerial images, but has the advantage that an explicit reconstruction of the local environment is not necessary. Many of the approaches discussed above filter a camera's pose over multiple frames. This undoubtedly improves localization accuracy, and could be applied with our approach as well. In the current paper, however, we explore only what is possible when our approach is applied to a single frame.

III. Approach

The location of the camera in the urban terrain is determined by estimating, from a *single* omnidirectional query image, the footprints of visible building facades and then registering this local footprint data to the terrain model. Our approach consists of off-line terrain model building and on-line camera localization. There are two main steps in off-line terrain modeling: (1) Generating a 2D map from an aerial image, and (2) Computing FPO descriptors at all grid points in the map. On-line camera localization consists of three main steps: (1) Detecting building facades in the

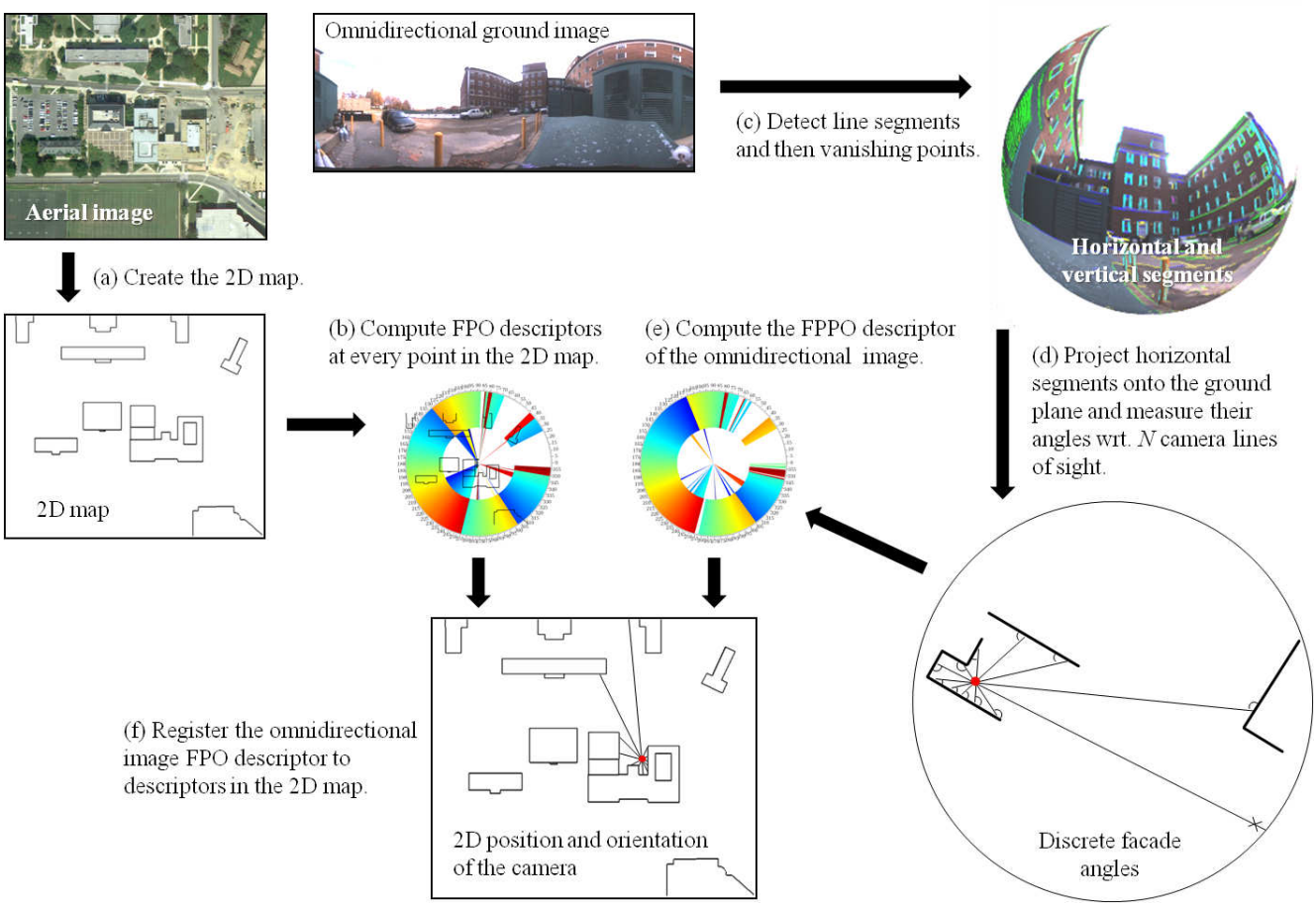

Fig. 1. Overview of our localization approach. A 2D map is manually created from an aerial image and footprint orientation (FPO) descriptors are computed at each point in this map. Given an omnidirectional ground image for an unknown camera location, line segments and vanishing points are detected in the image. Horizontal line segments are projected onto the ground plane and from these the query FPO descriptor is computed. The query FPO descriptor is registered to those of the 2D map to determine the position and orientation of the camera.

omnidirectional image; (2) Computing the FPO descriptor for the omnidirectional image; and (3) Searching for the FPO descriptor in the terrain model that is closest to the query FPO descriptor. These steps are illustrated in Figure 1 and each step is described in detail below.

A. Map Generation

Building boundaries are good features to use in modeling an urban environment because they can be detected in both aerial and ground imagery. Our terrain map is created from publicly available, high-resolution aerial imagery [23]. We process this imagery at a resolution of approximately 5 pixels/meter; this allows our localization system to obtain sub-meter position accuracy. Because the focus of our work is pose estimation, not terrain mapping, we create our map manually by outlining the boundaries of buildings in aerial images. For large urban environments, creating a terrain model by hand would be tedious. Fortunately, automated methods can be employed (*e.g.*, [21]) to convert an aerial image to a 2D terrain map. The 2D models that we create are not perfect for the following reasons. (1) Small errors are introduced by the manual outlining process; (2) The aerial images that we used are not true orthographic images, so there are small errors due to perspective; (3) The aerial images may be out of date; and, (4) We do not calibrate the aerial camera, so lens and sensor distortions may be present in the aerial images. As will be seen in Section IV, these errors don't appear to have a big impact on our system's accuracy.

Some buildings consist of wings of different shapes stacked on top of each other or butting up against each other. An example of such a building is shown in Figure 2. From a ground level view of the building, any combination of facades from these different wings may be visible to the camera. When a camera is close to a building, the lower section of a building is usually the only part that is visible. If the camera is further away, but there is an occluding object (*e.g.*, car or tree) between the camera and the building, then it's possible that only the upper section of the building will be visible. At any given horizontal viewing direction, multiple building facades with different orientations may be visible. The situation may also occur when a taller building lies behind a shorter building. This illustrates the need to model all major facades of a building, not just those on the outermost boundaries. Thus, when creating a map from an

Fig. 2. A complex building where multiple facades of different orientations may be visible in any single horizontal viewing direction.

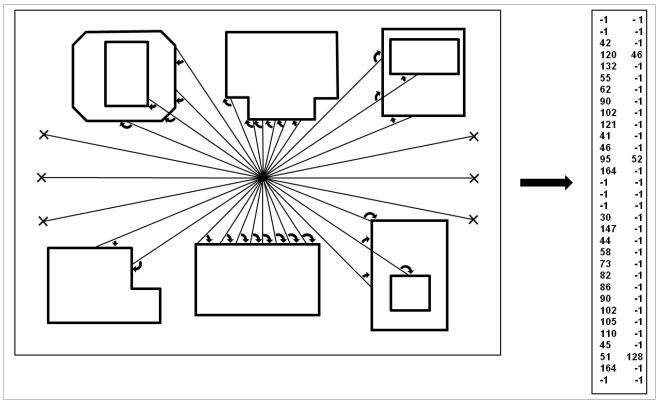

Fig. 3. The FPO descriptor is a $V \times D$ matrix of the angles of projected building facades measured at equally spaced viewing directions. Multiple facade angles may be measured for each viewing direction when $D > 1$. In this figure, $V = 32$ viewing directions and $D = 2$ facades, so every $360^{\circ}/32 = 11.25^{\circ}$ up to two different facade angles are measured.

aerial image, the user outlines, in addition to the outermost facades, internal facades on parts of buildings that rise up from the building below. Raised regions of a building can usually be identified from aerial imagery. Shadows will be found adjacent to raised regions when the aerial image was acquired on a sunny day; this makes identifying raised areas especially easy. The 2D map consists of a set of N building facades, $f_1, f_2, \ldots, f_N$, where each $f_i = \left(p_i^1, p_i^2, h_i\right)$ is a triple consisting of the 2D endpoints $\left(p_i^1, p_i^2\right)$ of the facade's projection onto the ground plane (the pixel coordinates in the aerial image), and the facades relative height h_i. The h_i say nothing about the true heights of building facades, which cannot be determined from an aerial image, but represent the *"higher than"* relationship for pairs of facades: facade f_i is higher than facade f_j if and only if $h_i > h_j$. The relative height values of building facades are used to generate the FPO descriptors. In cases where the user is unable to identify raised regions of buildings, the outermost facades by themselves provide an adequate description of the building as the outermost facades are the ones most frequently observed from the ground level.

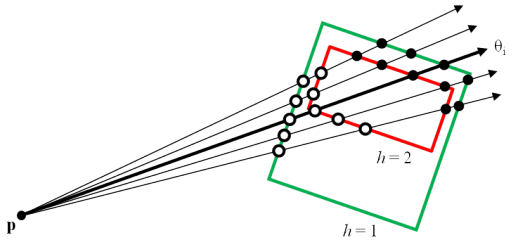

Fig. 4. Sampling the rays emanating from a point $\mathbf{p}$ in the direction θ_i. Segments on the green facade have relative height 1 and segments on the red facades have relative height 2. The white points contribute to the angles measured for direction θ_i of point $\mathbf{p}$, but the black points do not because their heights do not increase relative to points closer to $\mathbf{p}$; the black points are not visible because they are on the back side of the building. In this case, the angles associated with the white points will form two clusters, one cluster corresponding to the seven points on the two left-side facades and one cluster corresponding to the three points on the bottom red facade.

B. Orientation Descriptors for the Terrain Map

To perform localization, an offline process computes a FPO descriptors at all grid points in the 2D map and the online process computes one FPO descriptor for the omnidirectional query image. The FPO descriptor is a pair $\mathcal{F} = (\mathcal{R}, \mathcal{A})$ where $\mathcal{R}$ and $\mathcal{A}$ describe the relative and absolute orientations, respectively, of surfaces in the scene. As shown in Figure 3, the relative orientations, $\mathcal{R}$, is a set of samples of the angles between rays emanating from the camera (parallel to the ground plane) and the projection of the facades onto the ground plane. $\mathcal{R}$ is represented as a $V \times D$ matrix where V is the number of samples in the view-direction dimension and D is the maximum number of samples in the depth dimension. These angles are easily computed for any map point using simple geometry. The relative height values of facades determine which facades contribute to the samples for a particular viewing direction. In essence, the only facades that are visible are the ones that are taller than any closer facade: if $f_{i_1}, f_{i_2}, \ldots, f_{i_N}$ are N ordered facades that a ray emanating from a map point $\mathbf{p}$ in direction θ_i $(i = 1, \ldots, V)$ intersects with, then M of these facades, $f_{i_{j_1}}, f_{i_{j_2}}, \ldots, f_{i_{j_M}}$ $(j_1 < j_2 < \ldots < j_M)$, contribute to the angles measured along this ray provided the relative height values of these M facades are strictly increasing: $h_{i_{j_1}} < h_{i_{j_2}} < \ldots < h_{i_{j_M}}$. To make the descriptor less sensitive to small changes in orientation, we sample a few rays distributed around each θ_i: for direction θ_i from point $\mathbf{p}$ we sample facades on T rays uniformly distributed in the $360^{\circ}/V$ sector of the map centered at $\mathbf{p}$. This is illustrated in Figure 4 . These contributing angles are clustered and then the means of the D clusters with the most number of members are assigned to row i of $\mathcal{R}$. If there are fewer than D clusters, then the unassigned angles are set to -1. The absolute orientations, $\mathcal{A}$, of scene surfaces is a vector of the angles, relative to the camera's optical axis, of the visible surfaces projected onto the ground plane. Frequently, $\mathcal{A}$ consists of only two angles corresponding to two orthogonal orientations of the nearby visible building facades. In all experiments reported in this paper, we take $V = 360$, $D = 2$, and $T = 5$. Figure 5 shows the color-coded relative orientations computed for a particular map point.

An important question that arises is: when can the 2D pose

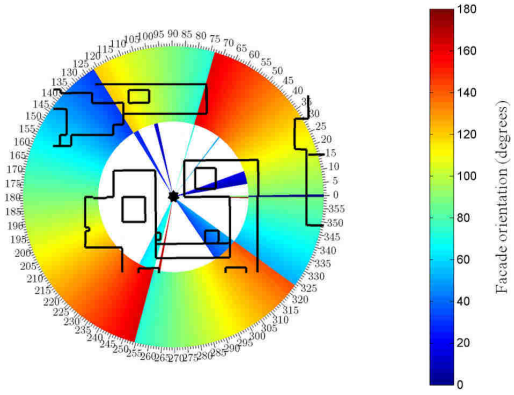

Fig. 5. The color-coded relative orientations computed for the map point at the star. At every viewing direction from 0 to 359°, up to two facade orientations (between 0 and 180°) are identified. Each ring represents a cluster of facade orientations for each of the 360 viewing directions. The entry for a viewing direction is white when no clusters are identified for that viewing direction. The inner ring only is white when only one cluster of orientations is identified.

of a camera be determined by sampling the orientations of a set of facades? Measuring in an image the orientation of exactly one facade in the map determines the orientation of the camera, but does not constrain its position. Measuring the orientation of a facade at a corner determines the camera's orientation and constrains its position to lie along a line emanating from the corner. When the location of a corner is not observed, but from the presence of other facades can be inferred to lie in a region of length δ along the facade, then the camera's position is constrained to lie in an infinite length linear strip of width δ emanating from the corner between of the two facades. Observing the locations of two or more corners determines two or more strips on which the camera must lie, and the position of the camera must lie at the intersection of these strips. Thus, our approach is unable to determine the position of a camera when one or fewer facade corners are visible in the image. This situation may occur when the camera is located in a long corridor with no visible building corners, or in cases of significant occlusion. Long corridors, which are rare in most urban environments, are better handled by appearance-based methods.

C. Facade Detection in Omnidirectional Images

To compute the FPO descriptor for an omnidirectional image, building facades in the image must be detected and their orientations estimated. Much previous research has explored the problem of building facade detection (*e.g.*, [24], [13], [25]). Most approaches designed to work with a single monocular camera assume that building facades contain edges corresponding to rooflines, doors, windows, bricks, etc., that are parallel or orthogonal to the ground plane. This is almost always a valid assumption in urban environments. We make the same assumption. We do not actually detect building facades, but only the edges that are likely to lie on facades. These edges do not need to be grouped into higher-level structures, we need only estimate their 3-space orientations, which is relatively easy to do when there are a sufficient number of parallel edges in the scene. Any individual facade may have only a small number of edges on it as long as there are enough edges parallel to it elsewhere in the scene. Determining the 3-space orientation of individual edges is a much simpler problem than segmenting or reconstructing planar facades; this enables our approach to achieve a level of robustness not found in previous 2D map-based visual localization methods.

Although the figures in this paper show 360° panoramas for the omnidirectional camera images, our omnidirectional image actually consists of a set of calibrated planar images. These may be generated by a catadioptric omnidirectional camera [26] or by panning a normal field of view camera about its focal point. Each of these planar images is processed as follows. The Canny edge detector [27] is used to generate a binary image of edge points. Straight line segments are then extracted from this edge image by linking edges into contours and then splitting the contours into nearly straight segments [28]. Short segments are discarded and then nonlinear line fitting is used to estimate the line parameters that minimize the sum of squared distances to the contour points.

Due to the effects of perspective projection, parallel edges in a scene will intersect at a common point when projected into an image [29]. This point, called the vanishing point, may be a finite or infinite point on a planar image, but it is always finite on a spherical image. Line segment clustering based on common vanishing points is used to detect sets of parallel edges and determine their 3-space orientations relative to the camera coordinate system. First, all image line segments are represented by the planes that pass through the origin of the omnidirectional camera and the 2D line segments. This representation allows estimation of the vanishing points from all planar images simultaneous; this is more robust than computing the vanishing points for each image separately. Any two noncoincident planes that pass through the origin intersect in a line through the origin. This line defines the vanishing point (and 3-space orientation) of the two original line segments. To compute the vanishing points, we use a standard technique based on RANSAC [30], [29]. Pairs of planes are randomly selected, their intersection is computed, and support for that vanishing point is tallied. This is repeated a fixed number of times and then the vanishing point with the most support is selected. Nonlinear least squares is then used to estimate the optimal vanishing point using all planes of support. These planes are removed from further consideration and the process is repeated. The process ends when the best new vanishing point has insufficient support, or five vanishing points have been identified. A final step is to disambiguate the orientation of line segments that align with multiple vanishing points. These ambiguous segments are assigned the orientation of the most common compatible vanishing point from among its nearest neighbors. Vanishing points give the orientation of image segments, but not their distance from the camera. We end up with a set of 3D line segments, each defined up to an unknown scale factor. Figure 6 shows a panoramic

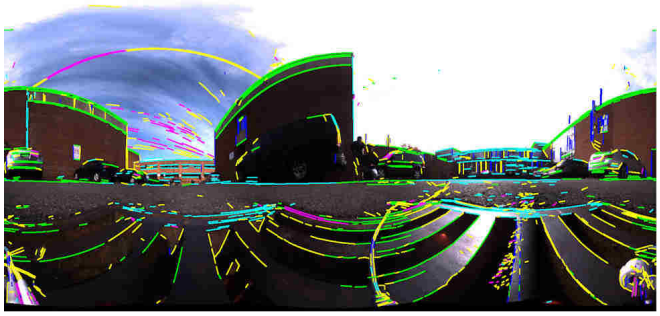

Fig. 6. An omnidirectional image (acquired from the position shown in Figure 5) and detected image line segments color-coded according to their 3D orientation. Yellow line segments do not align with any vanishing point. All other colors correspond to a known vanishing point and 3D orientation.

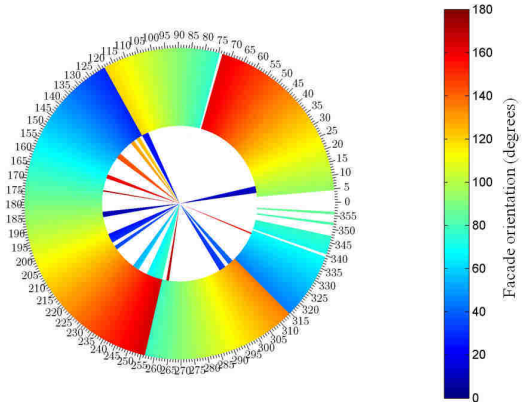

Fig. 7. FPO descriptor computed from the omnidirectional image shown in Figure 6.

image, acquired from the position shown in Figure 5, with line segments color-coded according to their 3D orientation.

D. Orientation Descriptors for Omnidirectional Images

As mentioned above, we assume that the edges on building facades are either parallel or orthogonal to the ground plane. This strictly isn't necessary for our approach; as long as the edges on a facade are parallel to each other, the orientation of that facade can be estimated from its vanishing point. However, the original assumption is almost always satisfied, and it allows us to discard a large amount of clutter due to vegetation, vehicles, etc. To compute the FPO descriptor for an omnidirectional image, we must project 3D line segments onto a horizontal ground plane and then measure the angles as described in Section III-B. We assume that the vertical axis in the scene corresponds to the image vanishing point that is closest to the camera's vertical axis. This assumption will be true provided the angle between the camera's vertical axis and the ground plane normal is less than $45°$. Before projecting onto the ground plane, to reduce clutter, we filter the 3D line segments by discarding those segments that are not close to parallel to the ground plane and those that lie within a fixed angle (typically $2°$) of the ground plane as seen from the omnidirectional camera. The later constraint removes surface markings on the ground plane. The remaining 3D line segments are projected onto a plane orthogonal to the vertical vanishing point. The result is a local 2D map similar to the urban terrain map except the ranges and relative heights of edges in the local map are unknown. Range is unnecessary for computing the FPO descriptor. Because all segments in the local 2D map are visible by the camera, the descriptor is computed as described in Section III-B but without enforcing the strictly increasing height constraint. Figure 7 shows the descriptor computed from the omnidirectional image shown in Figure 6.

E. Registering Map and Omnidirectional Image Orientation Descriptors

The final step in localization is to register the omnidirectional image FPO descriptor, $\mathcal{F}^i = (\mathcal{R}^i, \mathcal{A}^i)$, to 2D terrain map FPO descriptors, $\mathcal{F}^m = (\mathcal{R}^m, \mathcal{A}^m)$. The registration is performed by finding the map descriptor whose relative surface angles are closest to those of the image descriptor. In general, this distance must be minimized over all possible rotations of the image descriptor. For each viewing direction $r \in \{1, 2, \ldots, V\}$ and rotation $\omega \in \{0, 1, \ldots, V-1\}$[1], only the minimum distance pair of relative surface angles (one from the map descriptor and one from the corresponding row of the rotated image descriptor) contributes to the total distance; the remaining surface angles are due to complex building shapes or clutter and are ignored. To provide the registration process with additional robustness to clutter and occlusion, the cases when surfaces are observed in a direction of the map descriptor but not in the corresponding direction of the image descriptor, or vice versa, are handled differently. If a map descriptor records no surface angles in a particular direction, but the corresponding direction in the image descriptor does, then there is likely clutter in that direction of the scene if this correspondence is correct. Conversely, if an image descriptor records no surface angles in a particular direction, but the corresponding direction in the map descriptor does, then that part of the facade is either occluded or was not detected because it is too distant. In each of these "obvious clutter or occlusion" cases, the contribution to the total distance by that viewing direction is the average distance for viewing directions that do not fall into these special cases.

Let $\mathcal{R}_{r,c}$ denote row r (modulo V), column c of $\mathcal{R}$ and define $n(\mathcal{R}_r)$ to be the number of nonnegative angles in row r of $\mathcal{R}$. The case of obvious clutter or occlusion for a pair of viewing directions (r_1, r_2) is identified by the function $c(\mathcal{R}_{r_1}, \mathcal{R}'_{r_2}) = n(\mathcal{R}_{r_1}) > 0 \oplus n(\mathcal{R}'_{r_2}) > 0$ and, for a given rotation ω, the number of viewing directions that are not obvious clutter or occlusion is $q(\mathcal{R}, \mathcal{R}', \omega) = \sum_{r=1}^{V}(1 - c(\mathcal{R}_r, \mathcal{R}'_{r+\omega}))$. Then, the distance between $\mathcal{F}^m$ and $\mathcal{F}^i$ for rotation ω is

$$d(\mathcal{F}^m, \mathcal{F}^i, \omega) = d_1(\mathcal{R}^m, \mathcal{R}^i, \omega)\left(1 + \rho\left(\frac{V}{q(\mathcal{R}^m, \mathcal{R}^i, \omega)} - 1\right)\right) \quad (1)$$

[1]Index ω represents a rotation of $360 \times \omega / V°$.

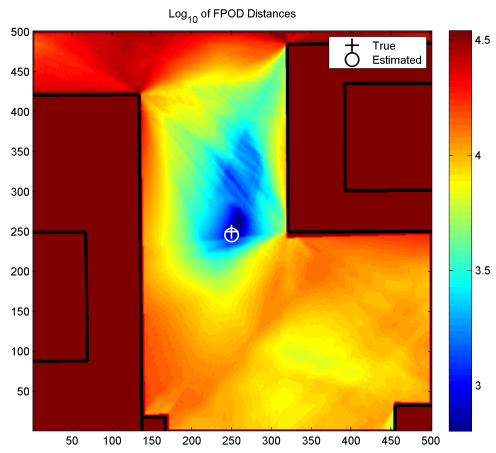

Fig. 8. Logarithm of the distances between the FPO descriptor computed for the image in Figure 6 and the FPO descriptors for the map shown in Figure 5.

where

$$d_1(\mathcal{R},\mathcal{R}',\omega)=\sum_{r=1}^{V}\left(1-c\left(\mathcal{R}_r,\mathcal{R}'_{r+\omega}\right)\right)d_2\left(\mathcal{R}_r,\mathcal{R}'_{r+\omega}\right) \quad (2)$$

and

$$d_2\left(\mathcal{R}_{r_1},\mathcal{R}'_{r_2}\right)=\min_{\substack{1\le s_1\le D\\ 1\le s_2\le D}}\left|\mathcal{R}_{r_1,s_1}-\mathcal{R}'_{r_2,s_2}\right|. \quad (3)$$

The parameter ρ in Eq. 1 determines the weight assigned to obvious clutter or occlusion relative to the nonobvious cases. We take $\rho = 1.5$ so that each obvious clutter or occlusion direction is assigned a distance that is 50% larger than the mean distance of the nonobvious cases.

Eq. 1 is minimized over all possible rotations ω of the image descriptor. However, the absolute angles, $\mathcal{A}^m$ and $\mathcal{A}^i$, of the visible surfaces in the map and image greatly constrain the relative rotations that need to be examined. For a typical environment where only two orthogonal vertical facade directions are visible, only four different orientations must be examined. In minimizing the distance between $\mathcal{F}^m$ and $\mathcal{F}^i$, the rotations that are examined are $\omega = \omega_1 + \omega_2$, where $\omega_1 \in \mathcal{A}^m$ and $\omega_2 \in \mathcal{A}^i$.

Figure 8 shows the logarithm of the distances between the FPO descriptor computed for the image in Figure 6 and the FPO descriptors for the map shown in Figure 5. As can be seen, the global minimum is very well defined and is very close (0.3 meters) to the camera's true location.

IV. Experiments

We use a Point Grey Research Ladybug 2 omnidirectional sensor. This sensor consists of six 1024×768 resolution cameras mounted in a single enclosure. The focal points of the six cameras are close enough that the six images can be considered to be rendered from a single point of view. Parallax from the different cameras is very small for objects more than a meter or two from the sensor. We collected 15 omnidirectional camera images in a variety of settings on a university campus. In order to determine the accuracy of our approach, we placed the camera in locations that are identifiable in the aerial image to an accuracy of around 0.25 m. These are places such as the corners of parking lot stripes or corners of sidewalks. The pose of the camera was determined by searching a $100\text{m} \times 100\text{m}$ region of the map centered at the known true location. The point in this region that minimizes the distance in Eq. 1 is the estimated position of the camera. We were unable to accurately measure the true orientation of the camera when collecting our data, so we don't report on the orientation accuracy. However, position and orientation are tightly coupled in the distance function, so if one is accurate, the other should be as well. The position errors for these 15 tests are shown in Table I, and some of the camera poses are shown in Figure 9. Our algorithm found a camera pose whenever two or more nonparallel building facades were observed (tests 1-12). In these cases, the mean position accuracy was 0.9 meters with standard deviation of 0.5 meters. Test 10, whose position error was 2.1 meters, was very challenging because the camera was located on the ground in a parking lot, and all visible buildings, which were far from the camera, were significantly occluded by parked vehicles. In tests 13-15, due to thick vegetation, only one side of a single building was detected in the omnidirectional image. For the reasons explained in Section III-B, our approach was unable to determine the position of the camera in these cases.

Test #	Position Error (m)
1	0.607
2	1.335
3	0.301
4	0.43915
5	0.86648
6	0.429
7	1.11902
8	0.6409
9	1.33
10	2.146
11	0.7528
12	1.1722
13	No pose
14	No pose
15	No pose

TABLE I
Camera position errors for localization tests.

V. Conclusions

We have developed the footprint orientation descriptor that provides a 2D geometric description of urban environments that is tolerant of significant clutter and occlusion. This descriptor, which is efficiently computed from an omnidirectional image, is matched to 2D maps to estimate the position and orientation of a camera. The use of 2D maps for localization in urban environments enables systems using our approach to be quickly deployed in new environments. Our initial experiments demonstrate an accuracy that, to our knowledge, is better than any other 2D map-based visual

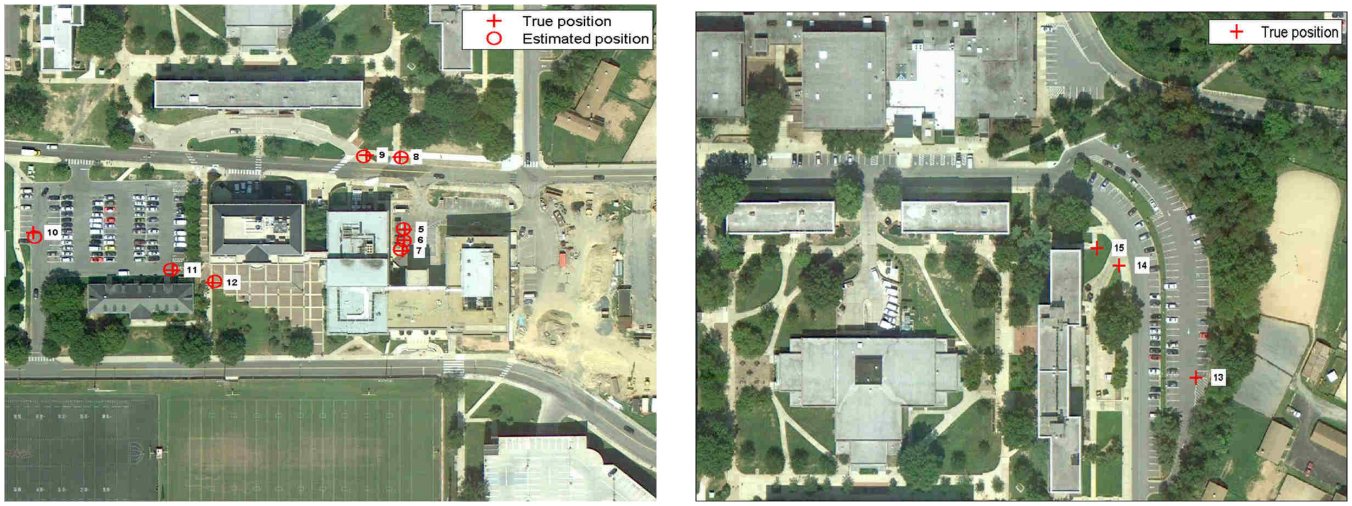

Fig. 9. True and estimated positions of the camera for tests 5-12 (left). Due to occlusions, no poses were found in tests 13-15 (right).

localization algorithm. In the future, we will conduct additional experiments in a wide variety of urban environments. We also plan to develop a real-time implementation of our approach and integrate it with other localization and navigation algorithms and sensors.

REFERENCES

[1] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proc. ICRA*, vol. 2, 2000, pp. 1023 –1029.

[2] T. Yeh, K. Tollmar, and T. Darrell, "Searching the web with mobile images for location recognition," in *Proc. CVPR*, 2004, pp. II–76 – II–81.

[3] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Proc. CVPR*, 2007.

[4] A. Murillo and J. Kosecka, "Experiments in place recognition using gist panoramas," in *Proc. ICCV, Workshops*, 272009-oct.4 2009, pp. 2196 –2203.

[5] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii state of the art," *Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108 –117, 2006.

[6] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, L. Murphy, W. Churchill, D. Cole, and I. Reid, "Navigating, recognizing and describing urban spaces with vision and lasers," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1406–1433, November/December 2009.

[7] S. Se, D. Lowe, and J. Little, "Vision-based global localization and mapping for mobile robots," *IEEE Trans. on Robotics*, vol. 21, no. 3, pp. 364–375, June 2005.

[8] D. Johns and G. Dudek, "Urban position estimation from one dimensional visual cues," in *The 3rd Canadian Conf. on Computer and Robot Vision*, 2006.

[9] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *Int. Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT*, Chapel Hill, North Carolina, 2006.

[10] A. Ascani, E. Frontoni, A. Mancini, and P. Zingaretti, "Robot localization using omnidirectional vision in large and dynamic outdoor environments," in *IEEE Int. Conf. on Mechtronic and Embedded Systems and Applications*, 2008, pp. 576 –581.

[11] C. Siagian and L. Itti, "Biologically inspired mobile robot vision localization," *IEEE Trans. on Robotics*, vol. 25, no. 4, pp. 861 –873, 2009.

[12] A. R. Zamir and M. Shah, "Accurate image localization based on google maps street view," in *Proc. ECCV*, 2010, pp. 255–268.

[13] J. Kosecka and W. Zhang, "Extraction, matching and pose recovery based on dominant rectangular structures," *CVIU*, vol. 100, no. 3, pp. 274–293, December 2005.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[15] R. Schuster, N. Ansari, and A. Bani-Hashemi, "Steering a robot with vanishing points," *IEEE Trans. RA*, vol. 9, no. 4, pp. 491–498, August 1993.

[16] J. Kosecka and W. Zhang, "Video compass," in *Proc. ECCV*, May 2002, pp. 657–673.

[17] B. Magnier, F. Comby, O. Strauss, J. Triboulet, and C. Demonceaux, "Highly specific pose estimation with a catadioptric omnidirectional camera," in *IEEE Int. Conf. on Imaging Systems and Techniques*, 2010, pp. 229 –233.

[18] S. Ramalingam, S. Bouaziz, P. Sturm, and M. Brand, "Skyline2gps: Localization in urban canyons using omni-skylines," in *Proc. IROS*, 2010, pp. 3816 –3823.

[19] S. C. Lee, S. K. Jung, and R. Nevatia, "Automatic pose estimation of complex 3d building models," in *WACV*, 2002, pp. 148 – 152.

[20] M. McHenry, Y. Cheng, and L. Matthies, "Vision-based localization in urban environments," in *Proc. SPIE, Unmanned Ground Vehicle Technology VII*, vol. 5804, 2005.

[21] K. Leung, C. Clark, and J. Huissoon, "Localization in urban environments by matching ground level video images with an aerial image," in *Proc. ICRA*, May 2008, pp. 551–556.

[22] T.-J. Cham, C. Arridhana, W.-C. Tan, M.-T. Pham, and L.-T. Chia, "Estimating camera pose from a single urban ground-view omnidirectional image and a 2d building outline map," in *Proc. CVPR*, San Francisco, California, 2010.

[23] Google, Inc., "http://maps.google.com." [Online]. Available: http://maps.google.com

[24] F. Schaffalitzky and A. Zisserman, "Planar grouping for automatic detection of vanishing lines and points," *Image and Vision Computing*, vol. 18, no. 9, pp. 647–658, June 2000.

[25] B. Micusik, H. Wildenauer, and M. Vincze, "Towards detection of orthogonal planes in monocular images of indoor environments," in *Proc. ICRA*, May 2008, pp. 999–1004.

[26] S. Nayar, "Catadioptric omnidirectional camera," in *Proc. CVPR*, June 1997, pp. 482 –488.

[27] J. Canny, "A computational approach to edge detection," *PMAI*, vol. 8, no. 6, November 1986.

[28] P. D. Kovesi, "MATLAB and Octave functions for computer vision and image processing," Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, 2000, available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[29] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, April 2004.

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *CACM*, vol. 24, no. 6, pp. 381–395, June 1981.