Computer Vision and Image Understanding 114 (2010) 274–285

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Horizon matching for localizing unordered panoramic images ☆

Damien Michel [a], Antonis A. Argyros [a,b], Manolis I.A. Lourakis [a,*]

[a] Institute of Computer Science, Foundation for Research and Technology – Hellas, N. Plastira 100, Vassilika Vouton, 700 13 Heraklion, Crete, Greece
[b] Department of Computer Science, University of Crete, 714 09 Heraklion, Crete, Greece

ARTICLE INFO

Article history:
Received 8 March 2008
Accepted 2 March 2009
Available online 17 March 2009

Keywords:
Panoramic vision
Camera pose estimation
Levenshtein distance
Circular string matching
Structure and motion estimation

ABSTRACT

There is currently an abundance of vision algorithms which, provided with a sequence of images that have been acquired from sufficiently close successive 3D locations, are capable of determining the relative positions of the viewpoints from which the images have been captured. However, very few of these algorithms can cope with unordered image sets. This paper presents an efficient method for recovering the position and orientation parameters corresponding to the viewpoints of a set of panoramic images for which no a priori order information is available, along with certain structure information regarding the imaged environment. The proposed approach assumes that all images have been acquired from a constant height above a planar ground and operates sequentially, employing the Levenshtein distance to deduce the spatial proximity of image viewpoints and thus determine the order in which images should be processed. The Levenshtein distance also provides matches between imaged points, from which their corresponding environment points can be reconstructed. Image matching with the aid of the Levenshtein distance forms the crux of an iterative process that alternates between image localization from multiple reconstructed points and point reconstruction from multiple image projections, until all views have been localized. Periodic refinement of the reconstruction with the aid of bundle adjustment, distributes the reconstruction error among images. The approach is demonstrated on several unordered sets of panoramic images obtained in indoor environments.

1. Introduction

Numerous omnidirectional camera designs, each with its own merits and characteristics, have been proposed in the literature. Compared to conventional perspective cameras, omnidirectional cameras offer a much wider field of view. Therefore, they facilitate capturing large portions of the environment with few images and without resorting to the use of movable gaze control mechanisms such as pan-tilt units. This property has obvious advantages for vision applications such as mosaicing, surveillance, telepresence, map building and localization, justifying the increased interest in omnidirectional vision. On the other hand, the attractive combination of increased environment coverage with low pixel bandwidth comes at the price of more difficult image matching due to the distortions caused by the intricacies of image formation and the reduced visual acuity due to the limited maximum resolution of contemporary imaging sensors.

This paper is concerned with the challenging problem of determining the relative positions and orientations of the viewpoints corresponding to a set of unordered central panoramic images, i.e. an image set for which no a priori proximity ordering information is available. The aforementioned problem is hereafter referred to as unordered panoramic image localization and arises naturally when, for example, dealing with distributed camera networks or vision-based mobile robot navigation (e.g. the so-called "loop-closing" [7] and "kidnapped robot" [4] problems). Image localization can be addressed in the framework of the fundamental structure and motion (SaM) estimation problem and benefits from the wide field of view offered by a panoramic camera. This is because environment features remain visible in large subsets of images and critical surfaces are less likely to cover the whole visual field. Existing research on SaM recovery has achieved a high level of sophistication and has produced impressive results. However, most of this research has approached the problem focusing on image sequences. The underlying assumption is that images that have been acquired close in time have viewpoints that are also close in space and, therefore, can be processed by repeatedly applying short baseline algorithms, either in a batch [5,30,12] or in an sequential mode [6,24,2,33,38]. Short baseline algorithms typically determine image matches with the aid of Harris corners [8] or Lowe's SIFT descriptors [19]. Occasionally, the simplifying assumption of a locally planar ground is made and pose is represented with three parameters, i.e. two for translation and one for rotation around

☆ This work was partially supported by the EU FP6-507752 NoE MUSCLE.
* Corresponding author. Fax: +30 2810 391601.
E-mail address: lourakis@ics.forth.gr (M.I.A. Lourakis).
URL: http://www.ics.forth.gr/~lourakis (M.I.A. Lourakis).

1077-3142/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2009.03.006

the vertical axis [33]. Often, sequential image localization is solved concurrently with the spatial mapping problem, thus the family of related methods are collectively referred to as *simultaneous localization and mapping* (SLAM) [2,15]. When applied to a set of unordered images which is the case dealt with in this paper, SaM estimation becomes more challenging since no prior linear ordering of images exists and, therefore, image matching has to cope with arbitrarily large baselines. Furthermore, a suitable order for processing images has to be determined automatically. For these reasons, there exist few approaches that deal with SaM estimation from unordered image sets, e.g. [35,37,17].

The so-called appearance-based methods [41,14,10,21] are among the earliest ones proposed for image localization tailored to unordered panoramic images. Prior to being used in a certain environment, appearance-based methods require that representative images of it are acquired and manually associated with location information. During operation, an input image is compared against all reference images. The location whose associated reference image best matches the input one according to photometric cues, is reported as that corresponding to the input image. Therefore, such methods yield coarse, qualitative location information that is often described as topological localization. The approach of Lamon et al. [14] is particularly interesting since it shares with our work the idea of matching images using strings of visual features. In [14], strings consist of symbols encoding coarse features such as vertical edges and color patches, whose ordering conforms to the relative ordering of the corresponding features in the panoramic image. Since in the context of the present work we are interested in accurate metric localization, appearance-based methods will not be discussed any further. Sagues et al. [34] borrow the idea of maintaining a database of reference views and rely on a set of images whose positions and orientations have been measured manually. Nevertheless, image similarity is assessed with a geometric procedure that relies on vertical line matching guided by the radial trifocal tensor to identify the reference image that is most similar to an unknown one. An unknown image is finally localized by computing its relative motion with respect to a pair of close reference images. Thus, the method is semi-automatic, requiring a fair amount of tedious manual localization of the reference images. More relevant to our work is the approach of Ishiguro et al. [9], who employ a set of cameras that have been placed at the same height and rely on moving objects to statistically determine the baselines of camera pairs, even when the two cameras are not visible from each other.

This work puts forward a novel approach for determining the relative locations and orientations of a set of unordered panoramic images, along with structure information in the form of a 2D map of their imaged surroundings. The main requirement is that all panoramic images must have been captured from the same height above a planar ground and with their optical axes perpendicular to it. Other than that, introduction of artificial markers or other modifications of the environment are avoided. The proposed approach operates sequentially, deducing the proximity of image viewpoints by employing the *Levenshtein distance* (LD) to compare circular strings derived from image data confined to horizons. The latter supply strong geometric cues related to the environment and have been previously employed for navigation of roving and flying robots in outdoors settings, e.g. [13,1]. Thus, the LD determines the order in which images should be processed and also provides matches among them, from which their corresponding environment points can be recovered. Environment points need not be visible in all images in order to be recovered. Recovered points that are visible in multiple images permit more images to be reconstructed through resectioning, which in turn allows the recovery of more points via triangulation and so on, until all images have been included into the reconstruction. Periodic refinement of the reconstruction with the aid of bundle adjustment, distributes the reconstruction errors among images. The proposed approach does not require any type of training. Furthermore, it has reasonable computational requirements, therefore is amenable to a near real-time implementation on ordinary hardware. An earlier version of the approach has appeared in [23]. Compared to that, the present description is more elaborate and encompasses a more rigorous and rapid horizon matching algorithm [36], an automatic mechanism for selecting the first two images that bootstrap the reconstruction, a more efficient mechanism of ranking panoramic images according to the spatial distance of their viewpoints and more detailed experimental results that include investigations of the method's performance for various resolutions of input images and in the presence of occlusions.

There are two major contributions from this work. First, it is shown that the LD, an established string distance metric, can be successfully applied to a fundamental problem in vision. Even at low resolutions, the LD is shown to be able to support the ordering of a set of images according to the spatial proximity of their viewpoints and to provide reliable matches among points on their horizons. Second, a method is proposed that relies on image horizons, which in essence are 1D images, to effectively register an unordered set of several panoramic images into a common coordinate frame without any knowledge of their relative positions or orientations and by not requiring the environment to be specially structured or contain predefined features. The proposed method is shown to be efficient and scalable, being capable of localizing several images of large spaces whose visual appearance changes considerably among viewpoints. Furthermore, and despite its use of a limited amount of image data, the method is demonstrated to be accurate and resilient to occlusions.

The rest of the paper is organized as follows. An overview of the proposed method is provided in Section 2. Section 3 concerns image matching using the LD and Section 4 deals with using the established matches for reconstruction from multiple panoramic images. Sample experimental results are reported in Section 5. The paper concludes with a brief discussion in Section 6.

2. Method overview

Assume that a set of images is available that has been acquired with a panoramic camera confined to move on a planar ground with its optical axis perpendicular to the former and at a constant height (cf. Fig. 3). It is desired to estimate the locations and orientations (i.e. pose) of the image viewpoints on a plane parallel to the ground, without any prior knowledge whatsoever of their relative spatial arrangement.

Panoramic images are associated on the basis of matched points. Under our assumed camera motion, the same planar "slice" of the environment is projected to the horizons of all images. Differently put, ground plane points at infinity project on the horizon of each panoramic image, i.e. a vanishing circle which is analogous to the vanishing line in a planar image. This implies that moving from one viewpoint to another causes horizon points to move along the horizon, but never away from it. This property is exploited to turn the 2D image matching problem into a 1D horizon matching one. More specifically, a string similarity measure is employed to compare the pixel strings corresponding to the horizons of the images to be matched, whereas pixels not on the horizons are ignored. The chosen string similarity measure is the Levenshtein distance [16], which corresponds to the minimum number of letter transformations that transform one string to the other and whose identification determines matches between string letters. Solving the correspondence problem via string matching was also proposed in [39] for the case of sparse corners. However,

in contrast to ours, that approach has several limitations such as the adoption of an approximate affine camera model, the assumption that intensity profiles lie on locally planar patches and the confinement of string matching to finding the longest common substring of two strings, without any provision for letter deletions and insertions due to corner detector failures.

Intuitively, images that have been acquired from nearby locations will have similar horizon pixel strings, therefore they will yield a low LD. On the other hand, the horizons of distant images will differ considerably, amounting to a large LD. Thus, the LD can (a) assess the proximity of the viewpoints of the compared images, coping even with wide baseline image pairs and (b) provide pixel correspondences between horizon pixel strings. Such a combination of properties is not attained by the various affine covariant region detectors compared in [25], which while being capable of providing matches between disparate views, cannot rank images according to the proximity of their viewpoints. Furthermore, and in contrast to scan matching techniques such as [3], computation of the LD does not require the relative pose of the two underlying images to be known beforehand. Since a panoramic horizon covers a 360° field of view, each pixel on it corresponds to an azimuth angle around the camera optical axis. Hence, pairs of matched horizon pixels allow the recovery of their corresponding environment points via triangulation. Image matching guides the reconstruction of image poses and the estimation of structure. A pair of images that share a large number of matches and a large baseline is selected first. This pair is used to recover an initial reconstruction. Then, the image that has the smallest LD with any of the reconstructed ones is added to the reconstruction by robustly estimating its pose from the known image to reconstructed point correspondences. This newly added image is used to reconstruct more points that are seen with sufficiently large viewing angles. Sparse bundle adjustment is used every few image insertions to jointly refine the motion and structure estimates through the minimization of the reprojection error. Details are elaborated in the following Sections 3 and 4.

3. Image matching

3.1. The levenshtein distance

The Levenshtein distance, also known as the *edit distance*, is a measure of the similarity between two strings of arbitrary lengths [16]. Given a pair of strings referred to as the source (s) and target (t), the LD corresponds to the minimum number of one-step edit operations (defined as letter deletions, insertions and substitutions), that are necessary to transform s into t. For example, for s = "VISION" and t = "VISITOR", $LD(s,t) = 2$ since two changes suffice to transform "VISION" to "VISITOR", i.e. inserting a "T" before the "O" and substituting "R" for "N". Note that, although not necessary, it was assumed that all edit operations are equiponderant with a cost of 1. The LD can be computed in $O(|s||t|)$ time by a dynamic programming technique, known as the Levenshtein algorithm. As a byproduct, this algorithm returns the pairs of letters that have been matched while computing the LD. The Levenshtein algorithm has the property of being order-preserving, that is retaining the order of matched letters. Thus, if a letter at position i in s matches the letter at position j in t, then letters in s at positions $k > i$ can only match letters in t that are at positions $l > j$. The LD has been employed in various domains in need of approximate pattern matching, such as spell checking, pattern recognition, speech recognition, information theory, cryptology, bioinformatics, etc. Regarding computer vision, use of the LD has been rather limited and has concerned the comparison of graph structures under edit operations, e.g. [28,42].

3.2. Horizon line matching

As already stated in Section 2, we assume color images acquired by a central panoramic camera confined to move at a constant height from a planar ground and with its optical axis perpendicular to it. A panoramic image can be unfolded with a polar-to-Cartesian isometry transformation that gives rise to a cylindrical image. Such an image is represented by a rectangular grid (cf. Fig. 1, top), whose vertical coordinates axis corresponds to a longitude that we will refer to as the image or viewpoint orientation. It can easily be verified that under the assumed motion model, the vanishing line of the ground plane corresponds to a straight horizontal line (i.e. a line of fixed y-intercept) in the unfolded image, which will hereafter be referred to as the *horizon line*. Moreover, the assumed camera motion guarantees that in the absence of occlusions, if an environment point projects on the horizon line of one view, then it appears on the horizon of any other view. Stated differently, the epipolar constraint for all points on the horizon of a panoramic image confines them to lie on the horizon line in any other panoramic image acquired under the assumed camera motion. Prior to extracting a horizon line, linear color normalization is performed separately to each color band to account for possible illumination changes. Furthermore, in order to allow for some tolerance in the case that the image plane of the panoramic camera is not exactly parallel to the ground, horizon lines are extracted through convolution with an 1D Gaussian filter of $\sigma = 2$, oriented vertically and centered on the line's expected location.

Considering the effects camera motion has on the appearance of the image horizon, pure translation is expected to expand the areas around the focus of expansion giving rise to pixel insertions, shrink areas around the focus of contraction resulting in pixel deletions and shift pixels in other locations by an amount dependent on scene structure [27]. Pure rotation is expected to introduce a constant, horizontal shift to all horizon pixels. General motion will have a combined effect. Pixel substitutions are also expected because of illumination changes, occlusion effects, imaging deformations and noise. Before applying the LD to the comparison of strings consisting of horizon pixels, the costs incurred by each edit operation should be defined. In this work, pixel deletions and insertions are assumed to have unit cost. The cost of a pixel substitution depends on the absolute differences of the RGB components of the pixels being compared. If any of these differences exceeds a certain threshold T, the substitution is assigned a fixed cost of two. Otherwise, the cost of substitution increases proportionally with the sum of the three cubed differences and assumes values in the range $[0,2]$. More precisely, the substitution cost for two color pixels with RGB color components (r_1,g_1,b_1) and (r_2,g_2,b_2) and whose color absolute differences are denoted by $\Delta r = |r_1 - r_2|$, $\Delta g = |g_1 - g_2|$ and $\Delta b = |b_1 - b_2|$, is defined as

$$S(\Delta r,\Delta g,\Delta b) = \begin{cases} \frac{2}{3\cdot T^3}(\Delta r^3 + \Delta g^3 + \Delta b^3) & \text{if } \max\{\Delta r,\Delta g,\Delta b\} \leqslant T, \\ 2 & \text{otherwise.} \end{cases} \quad (1)$$

The definition in Eq. (1) allows for some smoothness in the cost of substitutions and assigns low values when replacing pixels whose values differ slightly due to image noise and quantization effects. A value of 25 for T has produced good results in practice.

The LD involves the comparison of linear strings that have certain first and last letters. Panoramic horizons, however, are inherently cyclic and their origins in cylindrical images are arbitrary. Had the relative orientations of image viewpoints been known, this could have been remedied by circularly rotating all horizon strings so that their origins corresponded to the same absolute direction. Since the proposed approach does not make any assumption on the relative poses of panoramic views, the LD should be substituted

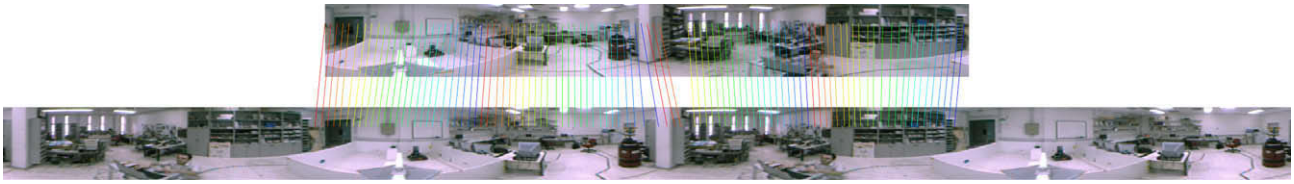

Fig. 1. Example of matches obtained by minimizing the LD between two horizon lines. Note that the bottom image is repeated twice. To improve readability, only one every 12 matches is shown and line segments of different colors are drawn between neighboring matching pixels.

by the *cyclic* LD that can account for the arbitrary linearization of horizons extracted from unfolded panoramic images. More specifically, the cyclic LD for two strings s and t is defined as $\min_{r\in\mathscr{P}(t)} LD(s,r)$, where $\mathscr{P}(t)$ denotes the set of all circular permutations of string t. The problem of cyclic sequence matching has attracted considerable interest and several algorithms have been proposed for computing the cyclic LD in less than the $O(|s||t|^2)$ time required by the trivial brute force algorithm comprised of computing the LD between s and every circular permutation of t. These algorithms are based on the observation that the cyclic LD can be computed as the minimum LD between s and any substring of length $|t|$ from $t\cdot t$, i.e. the concatenation of t with itself. Towards this end, they transform the problem into one of searching for minimum cost paths in a directed acyclic graph and employ dynamic programming techniques that prune the search space by exploiting previously computed paths. For instance, Maes [20] has developed a divide-and-conquer algorithm that has $O(|s||t|\log_2|t|)$ complexity, whereas Mollineda et al. [26] examine approximate algorithms. In [23], we have used a heuristic approach. Very recently, Schmidt et al. [36] proposed an algorithm to calculate the cyclic LD, which while having a worst case performance similar to that of [20], performs considerably faster in practice. Owing to its efficiency, the algorithm of [36] was adopted in this work to deal with computing the cyclic LD of horizon strings. For brevity, in the remainder of the paper LD will actually imply cyclic LD.

To match two horizon strings of equal length n, the algorithm of Schmidt et al. [36] works by first duplicating the target horizon string next to itself, thus ensuring that it can be matched with the source string without having to wrap around at string ends. Restricting matching to continuous substrings of length n from the duplicated target string excludes the possibility of matching both a target string pixel and its duplicate to different source pixels, a contingency that would violate the uniqueness stereo property. Fig. 1 provides two sample images of dimensions 1278×144 that were captured about 50 cm apart. Superimposed lines indicate some of the horizon pixel pairs matched between the two views as described above. Typically, the number of pixels matched between two images of this resolution is from 900 to 1000. It is worth pointing out that the order-preserving property of the Levenshtein algorithm that was mentioned in Section 3.1 also holds (in the cyclic sense) for cyclic matching. Therefore, the stereo ordering constraint is automatically enforced when matching horizon strings, ensuring that the order of matches is preserved along horizon lines.

4. Camera pose and scene structure estimation

4.1. Angular alignment of images

This section is concerned with estimating the relative rotation between two cylindrical images acquired at two different positions and whose optical centers lie on the plane $Z=0$. Note that this problem is more general compared to that of determining the relative orientation of two panoramic images acquired from the same spatial position, which can be effectively dealt with using the shift property of the Fourier transform as proposed in [29,22]. Fig. 2 shows two panoramic views at locations (X_c, Y_c) and (X'_c, Y'_c). At first, we are interested in recovering the angle θ that makes the two views parallel. Following this, we are also interested in recovering the angle ϕ that permits the alignment of both views with the direction of their relative translation.

Assume that the horizon lines of the image pair have been matched as explained in Section 3.2. Considering the disparities of horizon pixels, these have two components. The first, which varies from pixel to pixel, depends on the relative translation of the two images and the structure of the environment. The second depends on the relative rotation between the images and is the same for all pixels regardless of the environment. Thus, assuming that the average of positive translational disparities is approximately equal to the average of negative ones, the mean of all disparities approximates the disparity due to rotation. The assumption that positive and negative disparities cancel out boils down to an implicit assumption regarding scene structure. Nevertheless, experimental evidence indicates that this assumption is valid even in settings with considerable depth variations of no particular structure and in the presence of occlusions. The sought θ is simply the estimated circular shift that is necessary to align the second horizon line with the first.

Having canceled the rotation θ between the two images, the direction ϕ of the translational motion of one with respect to the other can be estimated based on the following observation. When a camera moves along a straight path without rotating, horizon pixels move so that positive and negative disparities define two half circles. These half circles are separated by the foci of expansion and contraction, which define the direction of translation [27]. Therefore, for two matched horizon lines with no relative rotation, the two antidiametric points separating the horizon pixels into two groups with opposite disparity signs yield the direction of translation ϕ as the direction of the line passing through them. As will shortly become clear, the angles θ and ϕ achieving the angular alignment of images are needed only when localizing an initial pair of reference images.

4.2. Localization and map building

Horizon line matching as described in Section 3 supplies the matched points required for image pose estimation and reconstruction. Let L denote the set of images that have been localized at some stage and U the set of those that remain to be localized. Initially, a pair of reference images is selected and L is initialized to containing them. More details on the choice of the reference images are given in the next paragraph. The origin of the coordinate system employed in the reconstruction is taken to coincide with the location of the first of the reference images. Then, the direction of translation of the second reference image with respect to the first is estimated as detailed in Section 4.1. This direction defines the angular coordinate of the second image; its radial coordinate is arbitrarily set to unity and corresponds to an unknown overall scale. After determining the relative positions of the two reference images, an initial map of the environment is recovered

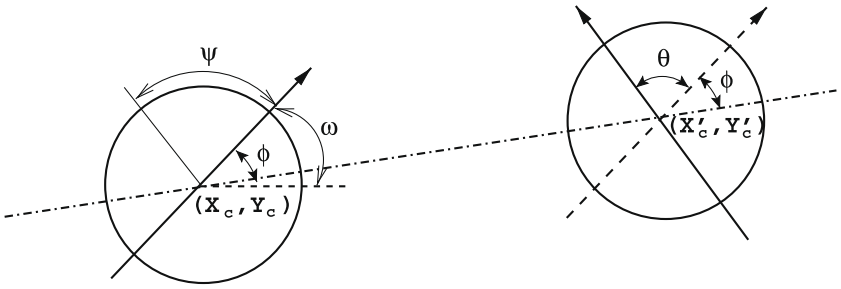

Fig. 2. The angles θ and ϕ define the relative orientation of two panoramic views. Angle ω defines the absolute orientation (viz bearing) of the left view in a global coordinate system and angle ψ corresponds to the azimuth angle to a horizon image pixel (see Section 4).

from them by triangulation from matched horizon points. More specifically, if an environment point is observed at an azimuth ψ by a camera at position (X_c, Y_c) with a bearing angle ω, its position (X, Y) on the plane parallel to the floor is constrained by (see also Fig. 2)

$$(Y - Y_c) - (X - X_c)\tan(\omega + \psi) = 0. \quad (2)$$

For two corresponding points in two images, Eq. (2) provides two linear constraints on (X, Y) from which the former can be determined.

We are now in the position to describe the strategy for selecting the two initial reference images. The first reference image is chosen arbitrarily. The second reference image completing the pair should be chosen so that a significant number of observed points can be reliably reconstructed from it. It is well known that the accuracy of point reconstruction via triangulation increases with the translational displacement between the employed images, i.e. their baseline. This is because a large baseline amounts to a large contained angle between the two backprojected rays originating at the image centers and, therefore, to a more precise estimation of their point of intersection. In this work, the second reference image is selected as the one being as close to the first as possible in terms of the LD and at the same time providing a large number of matches distributed all over the horizon line and seen with sufficiently large contained angles. These requirements translate to two images sharing many matches and having a baseline that gives rise to well-conditioned triangulation for most of them.

The availability of a map allows more images to be added to L through resectioning. More specifically, the image $I \in U$ that is closest to any of the images in L in terms of the LD is selected and removed from U. Being close to at least one of the reconstructed images ensures that I shares with it many points that have already been reconstructed. Thus, known map to image correspondences allow the location and pose of I to be estimated in a least squares manner from constraints on X_c, Y_c and ω arising from Eq. (2). Since this computation does not call for the estimation of the epipolar geometry, it is insensitive to the size of the baseline between I and its closest image from L, avoiding the frame selection problem that is common to many SaM algorithms [40]. To safeguard against errors arising mainly from mismatched points, the estimation of X_c, Y_c and ω is performed in a robust regression framework with the aid of the least median of squares (LMedS) estimator [32].

The process of directly comparing all cylindrical images in U against those in L for identifying I can be quite time-consuming, therefore an improvement over [23] consists in speeding it up by shortening the horizon strings to be matched, as follows. First, a coarse representation of each image is generated via repeated Gaussian smoothing and subsampling by a factor of two. Comparison of horizon lines is then performed at these coarse representations to determine I, i.e. the cylindrical image that is closest to one of those already reconstructed. Eventually, horizon pixel matches for I are determined by matching horizon strings at the original fine resolution. The resolution at which the horizons are compared might influence the order in which images are added to L, since the image I determined at a coarse resolution occasionally differs from the image J that would have been selected if the fine resolution was employed. Nevertheless, it has been verified experimentally that the shortest distances of both I and J from the images in L are very similar, resulting in the recovery of practically identical reconstructions upon termination. Yet another improvement to the performance of the horizons comparison process consists in caching the computed distances of images in U other than the chosen I to images in L so that they can be reused in the comparisons required during future iterations.

Once a new image I has been included in L, its matched points that are not already reconstructed can be added to the map. A point is reconstructed by examining all pairs of images in which its projections have been matched. Image pairs that give rise to small contained angles for the backprojected rays are removed from further consideration. A lower threshold of 15° is used to determine when the contained angle for a pair of rays is sufficiently large or not. Each of the remaining image pairs yields one estimate for the coordinates of the point to be reconstructed and the median of all such estimates provides a robust preliminary estimate. Finally, the point's coordinates are computed as the mean of the 70% of the estimates that are closer to the median one. To improve stability, a point is reconstructed only if it is visible in more than a minimum number of views, which is set to 7 in the current implementation. Following the introduction of new points in the map, the poses of all images in L are re-estimated in a robust fashion with the LMedS estimator[32]. Points that are marked as outliers in any of these estimations are removed from the map. Such points might be reconstructed again later if new constraints on their coordinates become available from the reconstruction of more images observing them. Each time a certain number of images (currently 5) have been added to the reconstruction, pose and structure estimates are simultaneously refined by minimizing their average image reprojection error through sparse bundle adjustment. Minimization of the mean reprojection error evenly distributes errors among reconstructed points and estimated image poses. Bundle adjustment was performed using our `sba` package [18]. The above steps are repeated until U becomes empty, i.e. all images have been localized.

At this point, it should be pointed out that an alternative approach for determining image poses would be to first estimate rotations as detailed in Section 4.1, then cancel their effect by derotation and finally use a reconstruction method such as the ones in [11,31,12] that assume pure translation and estimate the camera locations and environment structure in closed form. However, and despite their elegance, the batch operation mode of such methods precludes their use in cases where not all images to be

localized are available beforehand. Sample such cases, for example, are exploratory navigation scenarios in robotics and continuous pose estimation for augmented reality applications. Furthermore, batch translation-only techniques such as [11,31,12] have their own share of limitations, namely they require fairly long feature tracks to overcome problems with noise, minimize algebraic instead of physically meaningful geometric cost functions, have scalability problems with the number of available images and points, etc.

5. Experimental results

This section presents experimental evidence regarding the performance of a C++ implementation of the proposed method applied to cluttered indoor environments. The experiments reported in Sections 5.1–5.3 were carried out in a laboratory room, a CAD floorplan of which is shown in Fig. 7(a). Sample panoramic images from this room are illustrated in Fig. 4. The experiment of Section 5.4 used another room and its adjacent corridor. Sample images for that environment are shown in Fig. 11. The panoramic images utilized in all experiments were acquired with the aid of a central catadioptric color camera with a resolution of 640×480 pixels. This camera has a single effective viewpoint and is made up of an ordinary pinhole camera combined with a convex mirror. The y-intercept of horizon lines in the employed images was specified manually. The coarse resolution used for comparing the spatial distance corresponding to the viewpoints of horizon lines was 8 times lower than that of the input images, i.e. 80×60 pixels. The following subsections provide experimental evidence concerning different aspects of the proposed method.

5.1. LD compared to viewpoint distance

The first of the conducted experiments aims to verify a claim made in Section 2, namely that the magnitude of the LD depends upon the Euclidean distance between the viewpoints of the images being compared. To achieve this, a set of images whose pose can be determined fairly accurately was acquired as follows. The central catadioptric camera was rigidly attached to a rotating horizontal rod mounted on a vertical pole at a height of about 1.7 m above the floor (see Fig. 3). Rotating the rod with known angles effectively moved the camera along a circle. By varying the position of the camera along the rod and then completing a full revolution, more concentric circular trajectories were traced. In total, 48 images were captured, arranged on three concentric circles with radii 0.4, 0.9 and 1.4 m, each of which contained 16 images. For an arbitrarily chosen image on the inner circle, Fig. 5 plots the LD between it and every other image against the image locations that are within a square of side 3 m. To aid in visualization, a 3D surface interpolating the distances is drawn. This surface has a funnel-like shape, confirming that increasing Euclidean distances correspond to smoothly increasing the LDs. For future reference, the image set employed in this experiment will be referred to as "set 1".

Fig. 3. Imaging setup used for the experiments of Sections 5.1–5.3. The central catadioptric camera is attached to a rotating rigid rod, mounted horizontally on the vertical pole.

5.2. Accuracy of camera localization

The "set 1" images employed in the experiment of Section 5.1 were also used to quantify the accuracy of the camera poses estimated with the proposed method. Known circle radii and relative camera orientation angles provide the ground truth for the image poses, which can be compared to the camera pose estimated for each image by the proposed method. Fig. 6(a) facilitates the visual comparison of the estimated camera poses and the ground truth values. The estimated camera locations are shown with red circles while the true locations are shown with blue squares. Short lines on circles or squares indicate the orientations of the corresponding cameras. Clearly, the two sets of poses are in close agreement, as confirmed by the mean and standard deviation of the distance of the estimated camera locations from their true positions which are, respectively, 3.8 and 2.3 cm. The orientation error has a mean of 0.56° and a standard deviation of 0.98°. The 867 points reconstructed during localization are shown in Fig. 6(b). Note that no points lying on the walls that are far from the camera viewpoints are reconstructed, either because they are not seen by enough cameras with sufficiently large baselines or because they did not give rise to reliable image matches.

To test the method when the camera moves on a less regular trajectory, another experiment was conducted using an image set that will be denoted as "set 2". During this experiment, the camera was attached directly on the vertical pole which was then moved to 61 distinct positions covering most of the free space of the room. Application of the method to the acquired images recovered the camera positions and the map of reconstructed environment points shown in Fig. 7(b). A total of 2069 points were reconstructed. Circles are again used to represent the camera locations and short line segments the camera orientations. As can be seen by comparing this with the floorplan of Fig. 7(a), the layout and proportions of the room's walls have been reconstructed quite accurately, despite the presence of large textureless wall regions and significant variations of the amount of external light coming through the windows. It should be noted that the method has been able to reconstruct the pillar that exists in the middle of the room, overcoming ambiguities due to occlusions. The increased errors in the top left and right parts of the map are due to the lack of any texture on these areas of the walls that renders horizon matching more error-prone for them. No ground truth for the camera locations is available for this experiment due to the practical difficulties involved in measuring them in a global coordinate system. However, the distance of each camera location from its two nearest locations has been measured during image acquisition. Using the estimated camera locations, the mean and standard deviation of the distances to neighboring locations error were 1.8 and 1.7 cm, respectively. Overall, the reconstruction results are very satisfac-

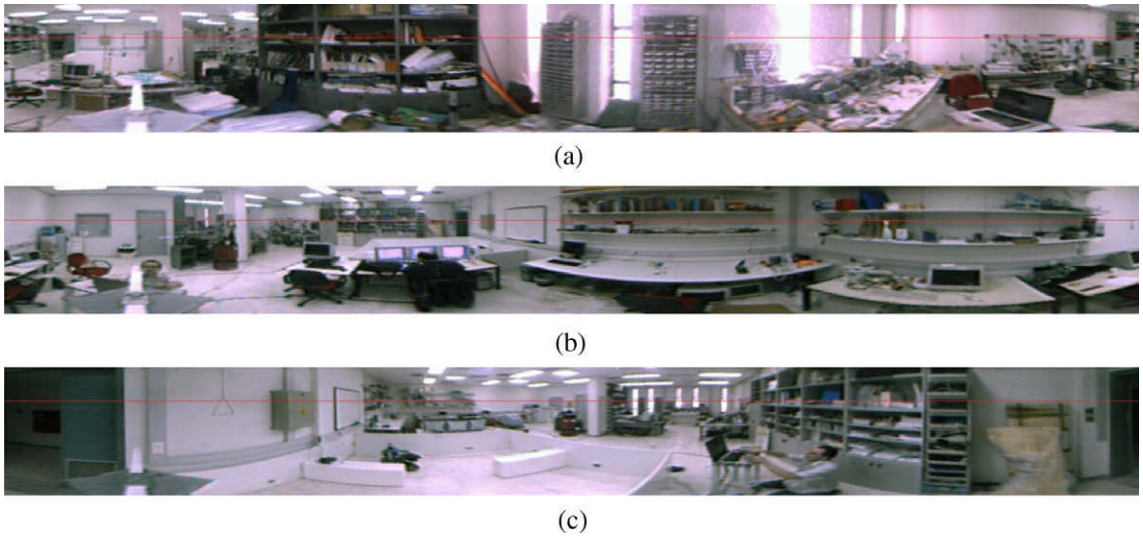

Fig. 4. Sample panoramic views of the room employed in the experiments of Sections 5.1–5.3. With reference to its floorplan shown in Fig. 7(a), image (a) is located in the right, (b) in the top left and (c) in the bottom left. The red line in each of these images corresponds to the location of the horizon.

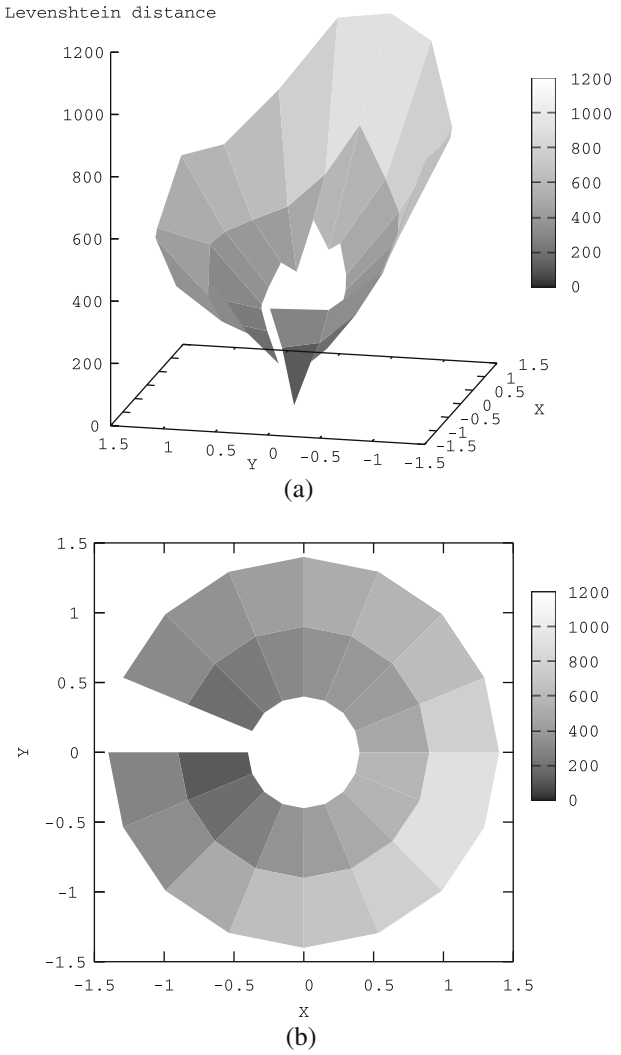

Fig. 5. Side (a) and top (b) views of the surface defined from the LD of image at location (−0.4,0) on the inner circle to all other images, plotted against the locations of image viewpoints.

tory, especially when considering the limited visual acuity of the employed camera. More specifically, the unfolded images are of dimensions 1278×144 pixels, which amount to approximately 3.5 pixels per degree. For comparison, assuming that the same imaging sensor was used for acquiring ordinary perspective images with a field of view equal to 50°, one degree would be imaged on 12.8 pixels. A video illustrating the progress of camera localization and structure estimation during this experiment can be found online at http://www.ics.forth.gr/~lourakis/panoloc.

The method has also been tested on the image set consisting of 109 views that resulted from the union of "set 1" and "set 2" employed in the previous experiments. It is worth mentioning that these two image sets have been acquired on different days that were about 2 weeks apart. Fig. 8 shows the camera poses and environment map recovered for the combined set. Computing the localization error in this case is not possible since the reference localization data for the two individual sets were measured independently, employing different coordinate frames. Nevertheless, and despite the larger number of images to be localized, the localization errors for the images of "set 1" and "set 2" within the combined set were found to be at the same level as those computed after separately localizing the individual sets. The number of reconstructed points totaled 2413. A related video illustrating the advance of localization is also available at http://www.ics.forth.gr/~lourakis/panoloc.

The three image sets employed for obtaining the reconstructions of Figs. 6–8 were reused in a set of experiments aimed at comparatively evaluating the performance of localization. More specifically, this set of experiments measured the processing times of three variants of the localization method, namely the one employing the heuristic algorithm for horizon matching that was described in our earlier description [23] (denoted in the following as "OMNIVIS"), the one employing the efficient matching algorithm of [36] at a single resolution (denoted as "ONERES") and that improving the former by performing matching at two resolutions as described in Section 4 (denoted as "TWORES" and being the one actually proposed in this paper). Furthermore, and in order to test the performance of the three variants with different resolutions for the input images, the former were applied to localizing the images resulting from repeatedly halving the original 640×480 ones by subsampling, down to a resolution

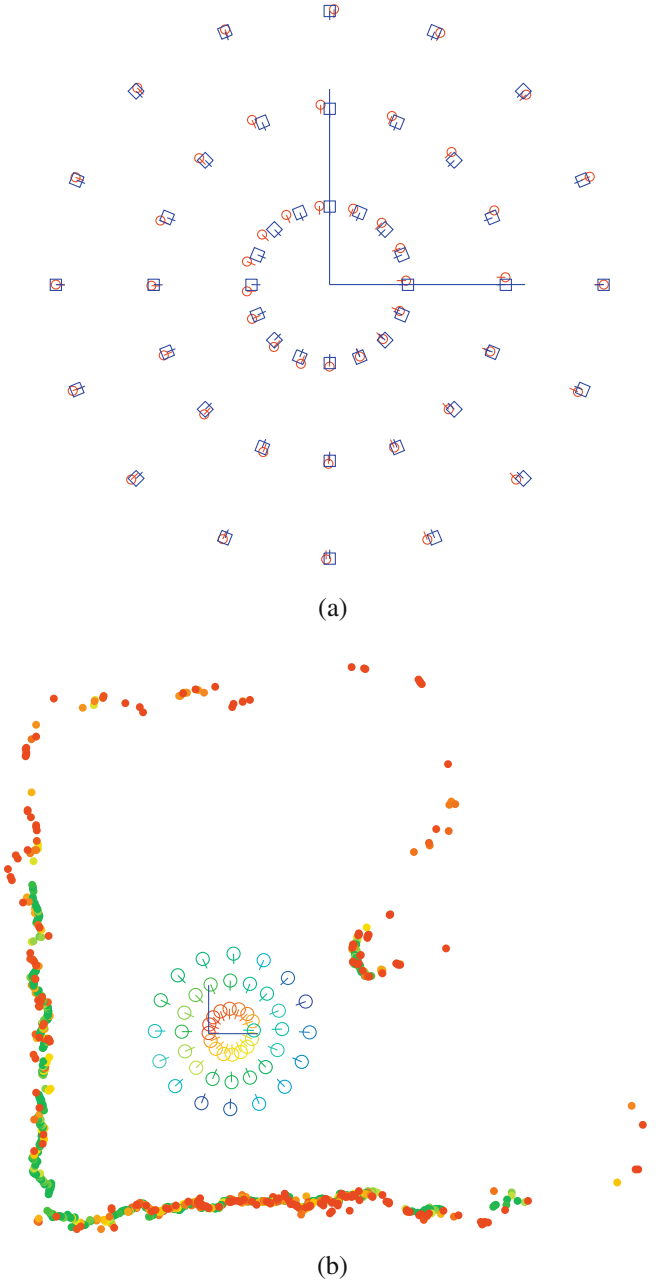

Fig. 6. (a) Estimated (circles) and true (squares) camera poses for a set consisting of 48 images captured with a camera moving on three concentric circles. (b) The reconstructed camera poses and environment points computed from the images of (a). The color of camera locations varies from red to blue in the order that the corresponding images were automatically selected for reconstruction. The color of reconstructed points varies from red to green according to the number of images from which they have been reconstructed. Red corresponds to points reconstructed from few images, green to those from many.

of 80 × 60 pixels. For each run corresponding to a certain resolution, the "OMNIVIS" and "ONERES" variants operate exclusively on images at that resolution. On the other hand, a run of the "TWORES" variant for some resolution employs the coarsest resolution (i.e. 80 × 60) for ranking images according to their spatial proximity, combined with the resolution at hand for determining horizon matches. The processing times for each image set, resolution and method variant are summarized in the bar graphs of Fig. 9. Each vertical bar corresponds to the sum of the total time spent for horizon matching (including the time required to rank images according to the LD) and the total time spent for recon-

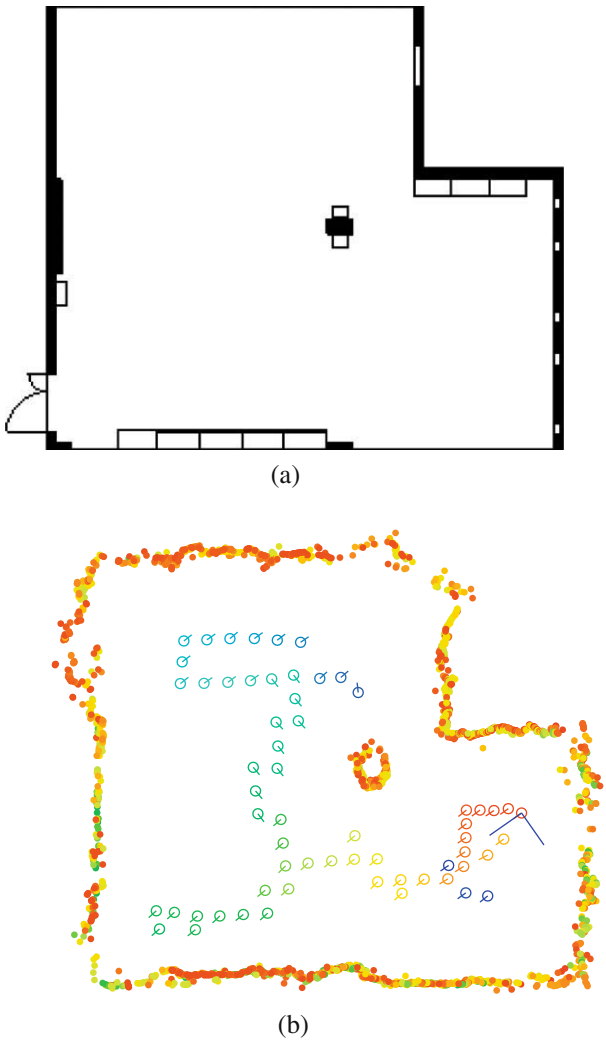

Fig. 7. (a) In scale floorplan of a laboratory room. The actual dimensions of the room are 9.4 × 10.5 m. (b) Reconstructed camera poses and environment points for the room of (a). The ratio of dimensions for the reconstructed room was 0.87, which compares favorably to a true value of 0.89 obtained from (a).

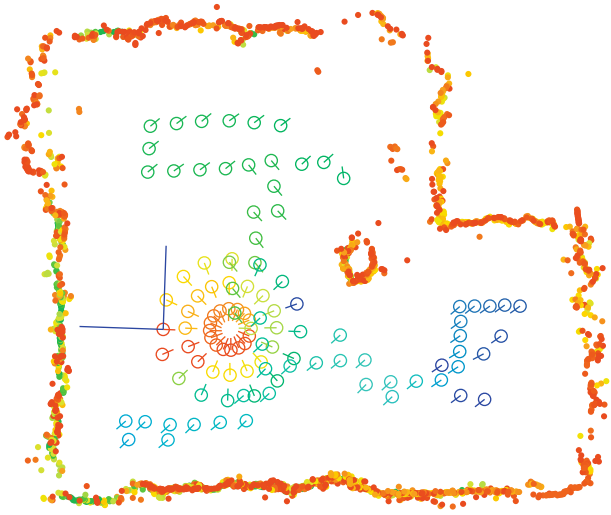

Fig. 8. Reconstructed camera poses and environment points when employing the two combined sets of images. Note that due to the choice of different initial reference images for the reconstruction, the scale and coordinate system origin differs from that of Fig. 7(b).

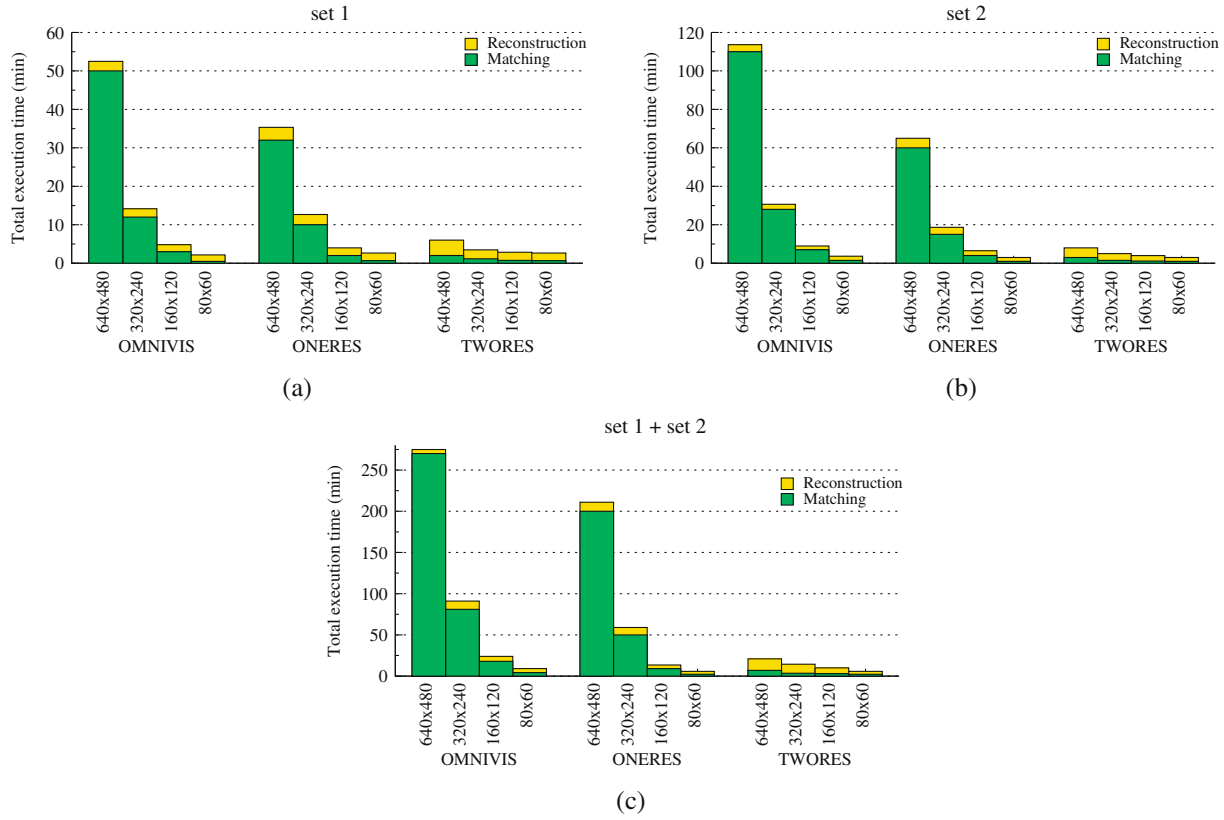

Fig. 9. Processing times obtained from three variants of the localization method applied to three image sets at four resolutions. Notice that the reconstruction times are very similar for all three localization variants.

struction. All timing measurements were collected on a laptop computer equipped with an AMD Turion 64 processor running at 1.6 GHz. As can be seen from Fig. 9, the processing times of "OMNIVIS" and "ONERES" are dominated by the time required for horizon matching. Adopting the faster matching algorithm of [36] in "ONERES" results in up to a twofold speedup over "OMNIVIS" for matching. However, even more dramatic performance improvements have been achieved by the combined use of a coarse and fine resolution by "TWORES". In this case, the time for matching is almost negligible and the time for reconstruction becomes the primary performance bottleneck. Overall, the "TWORES" variant was found to require on average a few seconds per input image. An indication of the influence of image resolution on the accuracy of localization is provided by the localization errors pertaining to the experiments of Fig. 9(a) and (b), which are included in Table 1. Evidently, the accuracy of the localization results produced by the three variants at all resolutions between the original one down to and including that of 160×120 is quite adequate. The small discrepancies observed in localization accuracy among the three methods are due to the different methods employed for computing the cyclic LD in each case. Specifically, "OMNIVIS" employs a heuristic, approximate matching method, whereas "ONERES" relies on an exact technique. Thus, given a certain horizon pair, slightly different matches might arise, which in turn yield slightly different reconstructed points. Compared to "ONERES", "TWORES" employs low-resolution, subsampled images to rank the horizon lines of not yet reconstructed images according to their proximity to the horizon lines of already reconstructed ones. During subsampling, some image data are lost and only low frequency texture information persists, thus the aforementioned ranking is determined from different inputs and can be slightly different between "ONERES" and "TWORES". Hence, these images might be included in the reconstruction in different orders, ending up in slightly different final maps.

Table 1
Mean and standard deviation for the localization error in centimeters and at various resolutions, computed for the experiments of Fig. 9(a) and (b).

Image set	Input resolution	OMNIVIS		ONERES		TWORES	
		Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
"set 1"	640×480	3.4	1.8	3.4	2.3	3.8	2.3
	320×240	5.0	3.0	6.0	3.6	6.0	3.6
	160×120	7.0	4.5	6.8	3.6	7.0	3.2
	80×60	46.0	21.0	16.0	9.0	16.0	9.0
"set 2"	640×480	1.9	1.9	1.9	1.9	1.6	1.6
	320×240	2.6	3.4	2.5	2.7	2.4	2.9
	160×120	4.0	3.7	4.0	3.9	4.2	5.5
	80×60	7.3	9.0	6.7	7.6	6.7	7.6

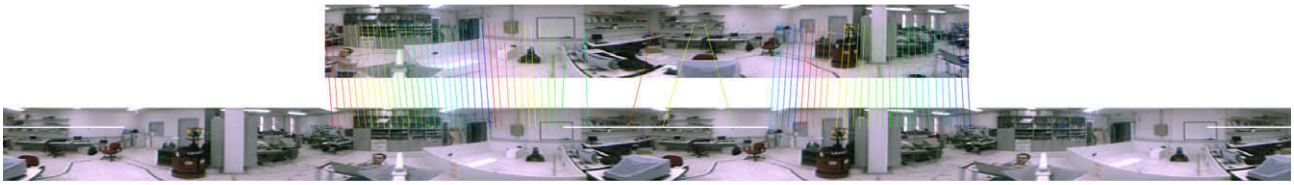

Fig. 10. Example of matches established between the horizons of two images in the presence of occlusion. In the bottom image, which is repeated twice, the horizon has been occluded in the area marked with the thick white line. Note that very few matches have been obtained in the occluded area.

5.3. Robustness under occlusions

A series of experiments focusing on the study of the robustness of image localization under occlusions is described next. Occlusions can be due to either the structure of the environment or transient moving objects such as people. The first case, also known as kinetic occlusion, can be effectively handled assuming that the available image matches provide sufficient environment coverage. In this study, we focus on the second type of occlusions which can give rise to image matches that do not conform to rigid environment structure and are, therefore, more difficult to handle.

For our purposes, occluding a cyclic horizon string is taken to amount to substituting a substring from it with arbitrary pixel values, not appearing in the same order in any of the remaining horizon lines (see also Fig. 10). Three are the factors describing the amount of occlusion in an image set: First, the fraction f of images that contain occlusions, second the size (i.e. subtended visual angle) of occlusions and third the color statistics of occluding pixels. To quantitatively assess the localization accuracy when systematically varying those factors, the following simulation approach was adopted. Initially, an image set is selected for which some prior reference data for the localization are available. Then, several sets with occlusion are generated synthetically from it by controlling the factors pertaining to occlusion: Images that will be occluded are selected randomly and in a manner ensuring that they amount to a fraction f of the total number of available images. The locations of occluded areas in the horizon strings of those images are determined by a uniformly distributed discrete random variable. The sizes of occluded areas are determined by constraining a random variable following a Gaussian distribution to lie within one standard deviation from its nonzero mean m. Color triplets for the occluding pixels are drawn from a trivariate Gaussian distribution whose mean and covariance matrix are estimated from the population consisting of the corresponding occluded pixels.

The proposed localization method (i.e. "TWORES") was applied to each of the generated sets with occlusions and the corresponding localization error was measured from the known reference data. The occlusion experiments reported here employed the 61 images in "set 2", with the fraction f varying from 10% to 100% and the sizes of occlusions being modeled by Gaussian distributions whose means m varied between 50 and 300 pixels (equivalent to a visual angle range between 14° and 85°) and their standard deviation equaled 10. Note that an occluded area around 300 pixels corresponds to an occluding object covering approximately one-fourth of the whole visual field. To ease the effect on the localization results of the exact choices made when synthesizing a particular image set, each experiment was run 20 times, each time using a different set of images occluded according to the chosen values for f and m. The average localization errors computed from all these runs are listed in Table 2. As can be clearly seen from them, the proposed method is remarkably resilient, being little affected by small and moderate amounts of occlusion. Likewise, the method manages to deliver reasonable location estimates even under severe occlusions. In this last case, it has been observed experimentally that the poorer performance of the method was due to the fact that occlusions prevented the projections of a significant fraction of environment points to be matched across the horizons of sufficiently many images. This, in turn, gave rise to the recovery of maps that were split into multiple parts which lacked global consistency despite satisfactorily capturing the local structure.

5.4. Application to an image sequence

The proposed method was also applied to an image sequence obtained in an environment different from the laboratory room depicted in the images of the previous experiments. This sequence was captured by mounting the panoramic camera on top of a mobile robot that moved smoothly along an L-shaped trajectory. More specifically, the robot started at one end of an oblong room, traversing it with constant velocity towards the opposite end on which the room's entrance is located. When the robot approached the entrance, it decelerated and exited the room into a corridor. Subsequently, the robot rotated in place for about 90° to align with the corridor and then moved along it. The baselines between successive images varied considerably and were around 30 cm when the robot moved with maximum velocity, decreased as it approached the door, became zero as the robot rotated in place and finally increased again up to roughly 30 cm as the robot moved along the corridor. The total distance traveled was about 16 m and a total of 71 images were acquired. A few selected images from the sequence are shown in Fig. 11. As can be observed, the appearance of the scene undergoes substantial changes along the followed trajectory. This is especially true during the transition from the room into the corridor, where most of the room's walls become occluded. Furthermore, large areas on the walls and especially at the level of the horizon, lack texture for providing strong matching cues. These factors combined with the fact that the camera undergoes small vibrations due to the robot's motion, contribute towards rendering the localization of the images in the sequence quite challenging.

These images were localized with the "TWORES" variant, treating them as an unordered set, i.e. without taking advantage of their known spatial order. Fig. 12 shows the reconstructed camera poses and environment points. As it can be confirmed from the colors assigned to camera locations that gradually change from red to blue, the order of reconstruction is identical to the natural order of the image sequence. The lack of ground truth measurements for the robot's trajectory precludes a quantitative evaluation for the performance of localization. Nevertheless, it is clear that the environment's ground plan has been adequately recovered and, as expected, the reconstructed camera locations are initially evenly separated, then become denser as the robot decelerates and then increase again in the corridor. A total of 2760 points were reconstructed and the running time required for the whole sequence was approximately 10 min. A video illustrating the progress of camera localization combined with the image sequence can be found online at http://www.ics.forth.gr/~lourakis/panoloc.

Table 2
Average localization error in centimeters corresponding to various amounts of occlusion in the images of "set 2". For comparison, the localization error in the absence of any occlusion is 1.6 cm.

Occlusion fraction f (%)	Subtended visual angle mean m in pixels (degrees)					
	50 (14.1)	100 (28.2)	150 (42.3)	200 (56.3)	250 (70.4)	300 (84.5)
10	1.88	1.90	2.03	1.86	2.36	4.85
30	2.01	2.19	2.75	3.78	6.08	108.38
50	1.98	2.14	4.46	6.59	12.75	106.44
70	2.23	2.67	4.88	32.08	60.20	90.52
100	2.09	3.17	11.38	45.94	50.35	97.67

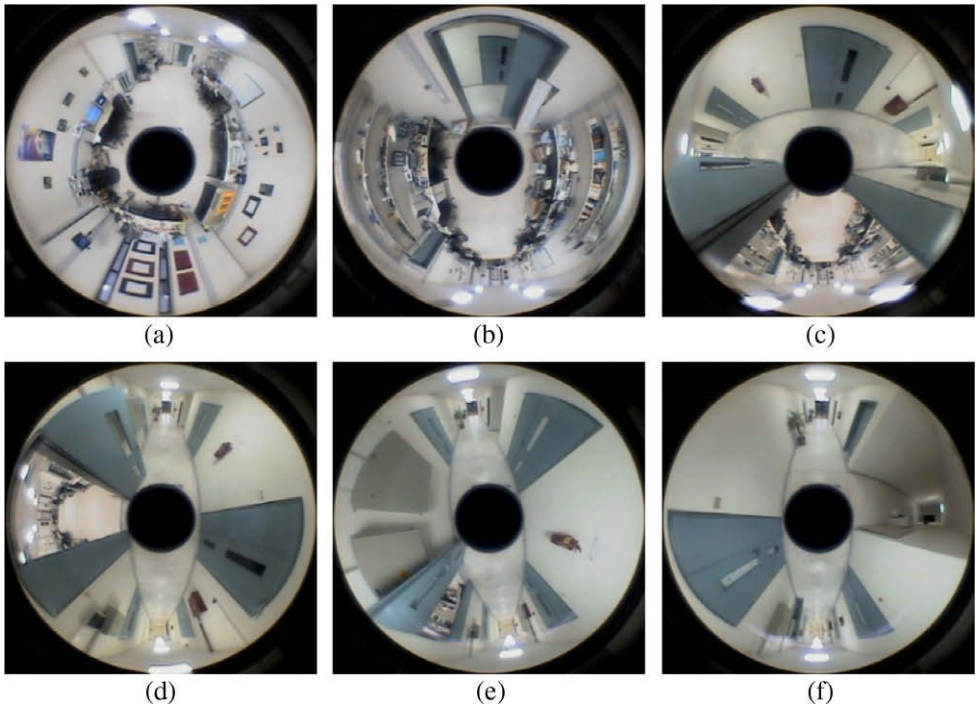

Fig. 11. Sample images from the sequence employed in the experiment of Section 5.4. The robot starts in one end of the room (a), moves towards the door (b), exits into the corridor (c), rotates to align with it (d) and moves along its center (e) and (f). Notice the very sparse texture of the corridor walls.

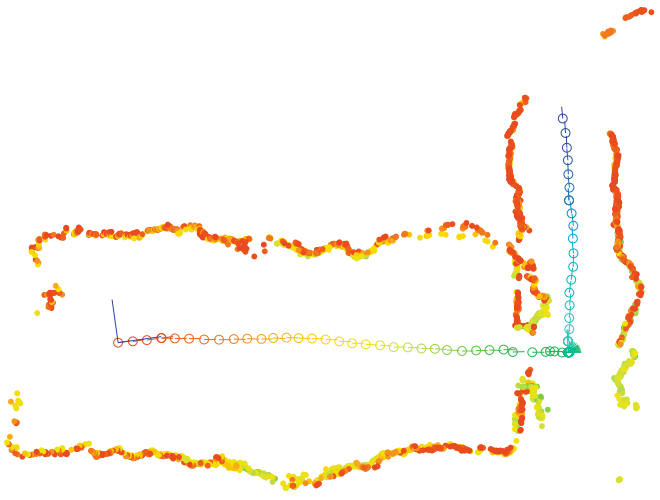

Fig. 12. Reconstructed camera poses and environment points for a robot moving in an oblong room and corridor. Notice that the reconstructed camera locations are initially evenly separated, then become denser as the robot decelerates and then increase again in the corridor. The "bump" in the recovered lower right corner of the room is caused by the protruding books on the shelf in that area that can be seen in the right middle part of Fig. 11(b). The physical dimensions of the room are 10.5×4.5 m and the corridor is 2.06 m wide.

6. Conclusion

This paper has presented an efficient and robust method for simultaneously localizing an unordered set of panoramic images and recovering a map of the environment. Matching a limited amount of image data confined to horizon lines has been shown to suffice for registering the images in a common coordinate frame and partially reconstructing the environment. The proposed method provides high quality results and can be employed to automatically determine the spatial arrangement of a set of panoramic cameras that comprise a distributed network, possibly being not observable from each other (i.e. invisible cameras in the terminology of [9]). In robotics terms, the proposed method works by perpetually solving the kidnapped robot problem, in which a mobile robot needs to be relocalized after having undergone an arbitrary motion that teleports it to some unknown location. Thus, it can be employed as a compact, low-cost, and odometry-free means for accurate optical positioning on moving platforms such as automated guided vehicles (AGVs) and mobile robots. Unlike techniques that rely on environment modifications in the form of buried wires, fluorescent paint strips or magnetic lines, our approach can determine position and orientation in two dimensions over wide, unmodified areas, thus enabling precise autonomous navigation, guidance and path following.

References

[1] J. Chahl, S. Thakoor, N.L. Bouffant, G. Stange, M.V. Srinivasan, B. Hine, S. Zornetzer, Bioinspired engineering of exploration systems: a horizon sensor/attitude reference system based on the Dragonfly Ocelli for Mars exploration applications, Robotic Systems 20 (1) (2003) 35–42.

[2] A. Davison, N. Molton, I. Reid, O. Stasse, MonoSLAM: real-time single camera SLAM, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 1052–1067.

[3] A. Diosi, L. Kleeman, Laser scan matching in polar coordinates with application to SLAM, in: Proc. of IROS'05, 2005.

[4] S. Engelson, D. McDermott, Error correction in mobile robot map learning, in: Proc. of ICRA'92, vol. 3, Nice, France, 1992.

[5] A. Fitzgibbon, A. Zisserman, Automatic camera recovery for closed or open image sequences, in: Proc. of ECCV'98, 1998.

[6] C. Geyer, K. Daniilidis, Structure and motion from uncalibrated catadioptric views, in: Proc. of CVPR'01, vol. 1, 2001.

[7] J.-S. Gutmann, K. Konolige, Incremental mapping of large cyclic environments, in: Proc. of CIRA'99, Monterey, CA, 1999.

[8] C. Harris, M. Stephens, A combined corner and edge detector, in: Proc. of the 4th Alvey Vision Conference, University of Manchester, UK, 1988.

[9] H. Ishiguro, T. Sogo, M. Barth, Baseline detection and localization for invisible omnidirectional cameras, International Journal of Computer Vision 58 (3) (2004) 209–226.

[10] M. Jogan, A. Leonardis, Robust localization using an omnidirectional appearance-based subspace model of environment, Robotics and Autonomous Systems 45 (1) (2003) 51–72.

[11] R. Kaucic, R. Hartley, N. Dano, Plane-based projective reconstruction, in: Proc. of ICCV'01, vol. I, 2001.

[12] J.-H. Kim, R. Hartley, Translation estimation from omnidirectional images, in: Proc. of DICTA'05, 2005.

[13] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, R. Wehner, A mobile robot employing insect strategies for navigation, Robotics and Autonomous Systems 30 (1–2) (2000) 39–64.

[14] P. Lamon, I. Nourbakhsh, B. Jensen, R. Siegwart, Deriving and matching image fingerprint sequences for mobile robot localization, in: Proc. of ICRA'01, vol. 2, 2001.

[15] T. Lemaire, C. Berger, I. Jung, S. Lacroix, Vision-based SLAM: stereo and monocular approaches, International Journal of Computer Vision 74 (3) (2007) 343–364.

[16] V. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics – Doklady 10 (8) (1966) 707–710.

[17] X. Li, C. Wu, C. Zach, S. Lazebnik, J.-M. Frahm, Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs, in: Proc. of ECCV'08, vol. 1, 2008.

[18] M.I.A. Lourakis, A.A. Argyros, SBA: a software package for generic sparse bundle adjustment, ACM Transactions on Mathematical Software 36 (1) (2009) 1–30.

[19] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[20] M. Maes, On a cyclic string-to-string correction problem, Information Processing Letters 35 (2) (1990) 73–78.

[21] E. Menegatti, T. Maeda, H. Ishiguro, Image-based memory for robot navigation using properties of omnidirectional images, Robotics and Autonomous Systems 47 (4) (2004) 251–267.

[22] E. Menegatti, M. Zoccarato, E. Pagello, H. Ishiguro, Image-based Monte Carlo localisation with omnidirectional images, Robotics and Autonomous Systems 48 (1) (2004) 17–30.

[23] D. Michel, A. Argyros, M. Lourakis, Localizing unordered panoramic images using the Levenshtein distance, in: Proc. of OMNIVIS'07, 2007.

[24] B. Micusik, T. Pajdla, Structure from motion with wide circular field of view cameras, IEEE Trans. PAMI 28 (7) (2006) 1135–1149.

[25] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L.V. Gool, A comparison of affine region detectors, International Journal of Computer Vision 65 (1–2) (2005) 43–72.

[26] R. Mollineda, E. Vidal, F. Casacuberta, Cyclic sequence alignments: approximate versus optimal techniques, Internation Journal of Pattern Recognition and Artificial Intelligence 16 (3) (2002) 291–299.

[27] R. Nelson, J. Aloimonos, Finding motion parameters from spherical flow fields (or the advantages of having eyes in the back of your head), Biological Cybernetics 58 (1988) 261–273.

[28] J. Ng, S. Gong, Learning intrinsic video content using Levenshtein distance in graph partitioning, Proc. of ECCV'02, vol. 4, Springer, Berlin, 2002.

[29] T. Pajdla, V. Hlavac, Zero phase representation of panoramic images for image based localization, in: Proc. of CAIP'99, 1999.

[30] M. Pollefeys, L.V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, Visual modeling with a hand-held camera, International Journal of Computer Vision 59 (3) (2004) 207–232.

[31] C. Rother, S. Carlsson, Linear multi view reconstruction and camera recovery using a reference plane, International Journal of Computer Vision 49 (2/3) (2002) 117–141.

[32] P. Rousseeuw, Least median of squares regression, Journal of the American Statistics Association 79 (1984) 871–880.

[33] E. Royer, M. Lhuillier, M. Dhome, J.-M. Lavest, Monocular vision for mobile robot localization and autonomous navigation, International Journal of Computer Vision 74 (3) (2007) 237–260.

[34] C. Sagues, A. Murillo, J. Guerrero, T. Goedeme, T. Tuytelaars, L.V. Gool, Localization with omnidirectional images using the radial trifocal tensor, in: Proc. of ICRA'06, 2006.

[35] F. Schaffalitzky, A. Zisserman, Multi-view matching for unordered image sets, or How do I organize my holiday snaps?, in: Proc. of ECCV'02, vol. 1, Springer, Berlin, 2002.

[36] F. Schmidt, D. Farin, D. Cremers, Fast matching of planar shapes in sub-cubic runtime, in: Proc. of ICCV'07, 2007.

[37] N. Snavely, S. Seitz, R. Szeliski, Photo tourism: exploring photo collections in 3D, ACM Trans. on Graphics 25 (3) (2006) 835–846.

[38] J.-P. Tardif, Y. Pavlidis, K. Daniilidis, Monocular visual odometry in urban environments using an omnidirectional camera, in: Proc. of IROS'08, 2008.

[39] D. Tell, S. Carlsson, Wide baseline point matching using affine invariants computed from intensity profiles, in: Proc. of ECCV'00, vol. 1, 2000.

[40] T. Thormaehlen, H. Broszio, A. Weissenfeld, Keyframe selection for camera motion and structure estimation from multiple views, in: Proc. of ECCV'04, vol. 1, 2004.

[41] I. Ulrich, I. Nourbakhsh, Appearance-based place recognition for topological localization, in: Proc. of ICRA'00, vol. 2, 2000.

[42] R. Wilson, E. Hancock, Levenshtein distance for graph spectral features, in: Proc. of ICPR'04, vol. 2, 2004.