# Localizing Unordered Panoramic Images Using the Levenshtein Distance*

Damien Michel, Antonis A. Argyros and Manolis I.A. Lourakis
Institute of Computer Science, Foundation for Research and Technology - Hellas
Vassilika Vouton, P.O. Box 1385, GR 711 10, Heraklion, Crete, GREECE
http://www.ics.forth.gr/cvrl

## Abstract

*This paper proposes a feature-based method for recovering the relative positions of 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 operates incrementally, 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 images, from which their underlying environment points can be recovered. Recovered points that are visible in multiple views permit the localization of more views which in turn allow the recovery of more points. The process repeats until all views have been localized. Periodic refinement of the reconstruction with the aid of bundle adjustment, distributes the reconstruction errors among images. The method is demonstrated on several unordered sets of panoramic images obtained in an indoor environment.*

## 1. Introduction

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 acquired. However, very few of these algorithms can cope with unordered image sets, i.e. image sets for which no a priori proximity ordering information is available. In this work, we are concerned with the problem of determining the relative positions and orientations of the viewpoints corresponding to a set of unordered central panoramic images. This 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" and "kidnapped robot" 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 a wide field of view facilitates 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. Furthermore, environment features remain visible in large subsets of images and critical surfaces are less likely to cover the whole visual field.

Most of the existing research on SaM recovery 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, e.g. [2, 1, 10, 7]. When applied to a set of unordered images, SaM estimation becomes more challenging since a suitable order for processing images has to be automatically determined. For this reason, there exist very few approaches that deal with SaM estimation from unordered image sets [13, 14]. The so-called appearance-based methods [4] 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. Thus, such methods yield coarse location information. Being interested in more accurate geometric localization, we will not discuss them any further. Sagues et al [12] 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. Image similarity, however, 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, involving a fair amount of tedious manual localization of the

*This work was partially supported by the EU FP6-507752 NoE MUSCLE.

reference images to function properly. More relevant to our work is the approach of Ishiguro et al [3], 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 paper puts forward a novel approach for determining the relative locations and orientations of a set of unordered panoramic images, along with certain structure information regarding their imaged surroundings. The proposed approach operates incrementally, deducing the proximity of image viewpoints by employing the *Levenshtein distance* (LD) to compare image data confined to horizons. Thus, the LD determines the order in which images should be processed and also provides matches among them, from which their underlying environment points can be recovered. Recovered points that are visible in multiple images permit the addition of more images to the reconstruction through resectioning, which in turn allows the recovery of more points 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.

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 matching problem in vision. Second, a method is proposed that relies on image horizons, which in essence are 1D images, to register an unordered set of several panoramic images into a common coordinate frame without any knowledge of their relative positions or orientations. This method is also shown to be scalable, being capable of localizing several images of large spaces whose visual appearance changes considerably among viewpoints. 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 it and at a constant height. 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.

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. 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; pixels not on the horizons are ignored. The chosen string similarity measure is the Levenshtein distance [5], which corresponds to the minimum number of letter transformations that transform one string to the other and whose computation determines matches between string letters. Solving the correspondence problem via string matching was also proposed in [15]. However, in contrast to ours, that approach has several limitations such as the adoption of an approximate affine camera model, the assumption that intensitity 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. Since a horizon covers a $360^\circ$ field of view, each pixel on it corresponds to an angle defining a bearing 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-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. Following sections 3 and 4 elaborate on details.

## 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 [5]. Given a pair of strings referred to as the source ($s$) and target ($t$), the LD corresponds to the minimum number of one-step 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 (i.e., insert 'T' before 'O' and substite 'R' for 'N') suffice. 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. A property of the Levenshtein algorithm whose importance will become clear in section 3.2, is that it preserves the order of matched letters. In other words, 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 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. [9].

### 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 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 image horizon, pure translation is expected to expand the areas around the focus of expansion (resulting in 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. 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 and imaging deformations. 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, 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 $[0, 2]$. A threshold value of 25 has produced good results in practice. The previous definition 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.

As defined, the LD compares 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 extended to account for the arbitrary linearization of horizons extracted from unfolded panoramic images. The problem of cyclic sequence matching has attracted considerable interest and several algorithms have been proposed for efficiently solving it [8]. For the purposes of this work, the technique described next has proved to perform well in practice.

The target horizon string is first duplicated next to itself, thus ensuring that it can be matched with the source string without having to wrap around at string ends. Nevertheless, target string duplication introduces a new problem, specifically the possibility that both a target string pixel and its duplicate are matched to different source pixels, thus violating the uniqueness stereo property. To deal with this problem, the source horizon string is repetitively matched with a substring of the duplicated target string that is aligned with the part of the latter yielding the most pixel matches and whose length is being progressively shrunk until it becomes equal to that of the source string. Figure 1 provides two sample images of dimensions $1278 \times 144$ that were captured about 50cm apart. Superimposed lines indicate 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 algo-

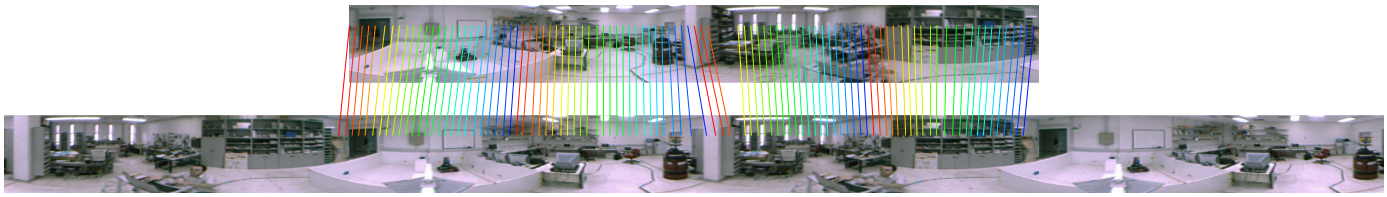


Figure 1. Example of matches obtained with the LD between two horizon lines. Note that the bottom image is repeated twice. To improve readability, only one every 12 matches is shown and line segments of different colors are drawn between neighboring matching pixels.

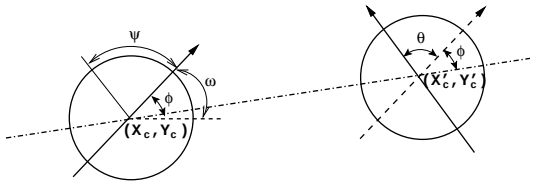



Figure 2. The angles $\theta$ and $\phi$ define the relative orientation of two panoramic views. Angle $\omega$ defines the absolute orientation of the left view in a global coordinate system and angle $\psi$ corresponds to the bearing angle of a horizon image pixel (see section 4).

rithm (section 3.1), automatically enforces the stereo ordering constraint when matching horizon strings, ensuring that the order of matches is preserved along horizon lines.

### 3.3. Angular Alignment of Images

This section is concerned with estimating the relative rotation between two cylindrical images whose optical centers lie on the plane $Z = 0$. Figure 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 $\theta$ that makes the two views parallel. Following this, we are also interested in recovering the angle $\phi$ 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. The disparities of horizon pixels 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 orientation 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. A circular shift of the second horizon line with the estimated rotation $\theta$ roughly aligns it with the first. Then, the shifted horizon line is rematched with the first using the Levenshtein algorithm and the aforementioned procedure is repeated for refining the estimated rotation. The procedure terminates when the change in the estimated rotation becomes too small.

Having canceled the rotation $\theta$ between the two images, the direction $\phi$ 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. 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 $\phi$ as the direction of the line passing through them. As will shortly become clear, the angles $\theta$ and $\phi$ achieving the angular alignment of images are needed only when localizing an initial pair of reference images. However, we have observed that the matches produced by the Levenshtein algorithm are of better quality when the images being matched are aligned. Therefore, prior to computing the final matches for two images, they are aligned by rotational shifting according to their estimated $\theta$ and $\phi$ angles.

## 4. Camera Pose and Structure Estimation

Horizon line matching as described in section 3 supplies the matched features 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. First, a pair of reference images is assumed to be manually identified and $L$ is initialized to containing them. These images should be selected so that they share a significant number of matches and, at the same time, have a large baseline so that estimating the structure of their observed points is well-conditioned. The origin of the coordinate system employed in the reconstruction is taken to coincide with one of the reference images. Then, the direction of translation of the remaining reference image with respect to the first is estimated as detailed in section 3.3.

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

$$(Y - Y_c) - (X - X_c)\tan(\omega + \psi) = 0. \qquad (1)$$

For two corresponding points in two images, Eq. (1) provides two linear constraints on $(X, Y)$ from which the former can be determined.

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 $\omega$ arising from Eq. (1). To safeguard against errors arising mainly from mismatched points, this computation is carried out with the aid of the LMedS robust estimator [11]. After $I$ has been included in $L$, its matched points that are not related to already reconstructed structure can be reconstructed and added to the map. The accuracy of point reconstruction via triangulation is known to increase 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, points are reconstructed by examining all pairs of images in which their 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 $15^\circ$ 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.

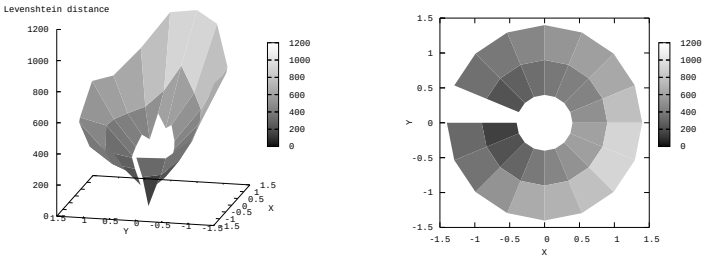
After new points have been introduced in the map, the poses of all images in $L$ are re-estimated in a robust fashion [11]. Points that are marked as outliers in any of these estimations are removed from the map. Such points might be reconstructed again later if new constraints on their coordinates become 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 the image reprojection error through sparse bundle adjustment. Minimization of the reprojection error evenly distributes errors among reconstructed points and image poses. Bundle adjustment was performed using our `sba` package [6]. The above steps are repeated until $U$ becomes empty, i.e. all images have been localized.

## 5. Experimental Results

This section presents experimental evidence regarding the performance of the proposed method in an indoor environment. Experiments were conducted with the aid of a central catadioptric color camera with a resolution of $640 \times 480$ pixels. This camera has a single effective viewpoint and is made up of a single perspective camera combined with a convex mirror. The y-intercept of horizon lines was specified manually. All reported experiments were conducted in a unmodified laboratory room, a CAD floorplan of which is shown in Fig. 5(a). Sample panoramic images of the room are also included in the supplementary material.

The first conducted experiment 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 panoramic camera was rigidly attached to a rotating horizontal rod mounted on a vertical pole at a height of about 1.7m above the floor. 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 could be traced. In total, 48 images were captured, arranged on three concentric circles with radii 0.4, 0.9 and 1.4 meters, each of which contained 16 images. For an arbitrarily chosen image on the inner circle, Fig. 3 plots the LD between it and every other image against the image locations that are within a square of side 3m. 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 increasing LDs.

The image set employed in the previous experiment was also used to quantitatively assess 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. Figure 4(a) facilitates the visual comparison

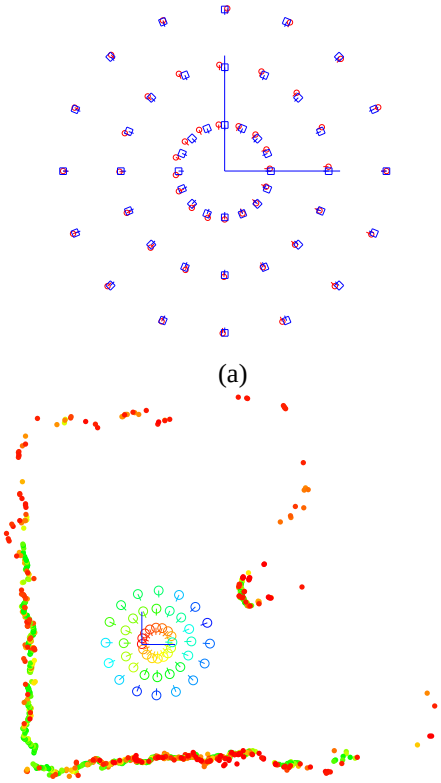



(a) (b)

Figure 3. 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 image viewpoint locations.

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 3.1cm and 1.5cm. The orientation error has a mean of $0.56°$ and a standard deviation of $0.98°$. The 898 points reconstructed during localization are shown in Fig. 4(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.

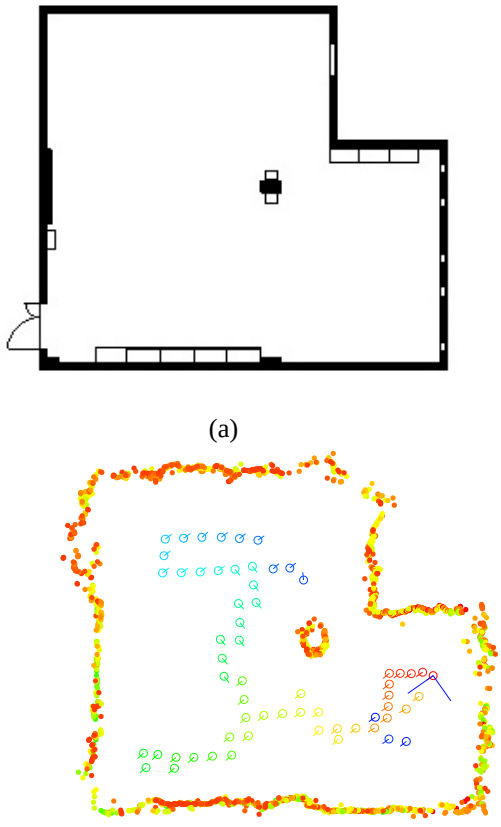
To test the method when the camera moves on a less regular trajectory, a third experiment was conducted. During this experiment, the camera was attached directly on the vertical pole which was moved to 61 positions covering much of the free space of the room. Application of the method to the acquired images recovered the camera positions and the map of reconstructed environment points shown in Fig. 5(b). A total of 2052 points were reconstructed. Circles are again used to represent the camera locations and lines the camera orientations. As can be seen by comparing this with the floorplan of Fig. 5(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 in 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 1.8cm and 1.9cm, respectively. Overall, the reconstruction results are very satisfactory, especially when considering the limited visual acuity of the employed camera. More specifically, the unfolded images were of dimensions $1278 \times 144$ 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 FOV of 50 degrees, one degree would be imaged on 12.8 pixels. A video illustrating the progress of camera localization and structure estimation during this experiment has been submitted as supplementary material.



(a)

(b)

Figure 4. (a) Estimated (circles) and ground truth (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 reconstructed. 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.



(a) (b)

Figure 5. (a) In scale floorplan of a laboratory room. The actual dimensions of the room are $9.4 \times 10.5$ meters. (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).

The method has also been tested on the image set resulting from the union of the two sets employed in the previous experiments and consisting of 109 views. Figure 6 shows the camera poses and environment map recovered in this case. The number of reconstructed points totaled 2450. Despite the larger number of images to be localized, the localization error for the combined image set was found to be at the same level as that for the individual sets. It is worth mentioning that these two image sets have been acquired on different days that were about two weeks apart.

## 6. Conclusion

This paper has presented a 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. Provided that the epipolar geometry is known, the proposed method can be readily extended to matching corresponding epipolar curves, thus covering the whole available visual field.

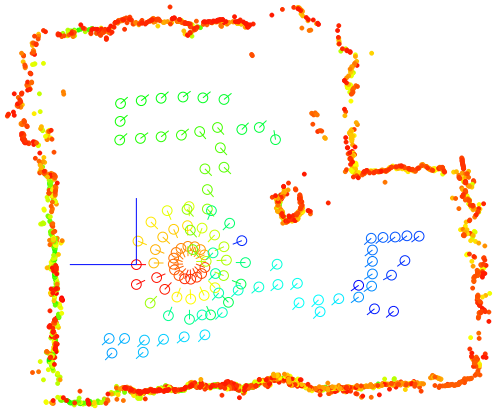


Figure 6. Reconstructed camera poses and environment points when employing the two combined sets of images. Note that due to the choice of different initial reference images for the reconstruction, the scale and coordinate system origin differs from that of Fig. 5(b).

## References


[1] A. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *Proc. of ICCV'03*, pages 1403–1410, Oct. 2003. 1

[2] A. Fitzgibbon and A. Zisserman. Automatic Camera Recovery for Closed or Open Image Sequences. In *Proceedings of ECCV'98*, pages 311–326, 1998. 1

[3] H. Ishiguro, T. Sogo, and M. Barth. Baseline Detection and Localization for Invisible Omnidirectional Cameras. *IJCV*, 58(3):209–226, Jul./Aug. 2004. 2

[4] M. Jogan and A. Leonardis. Robust Localization using an Omnidirectional Appearance-based Subspace Model of Environment. *Robotics and Autonomous Systems*, 45(1):51–72, 2003. 1

[5] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics - Doklady*, 10(8):707–710, Feb. 1966. 2

[6] M. Lourakis and A. Argyros. The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm. Technical Report 340, ICS-FORTH, Aug. 2004. 5

[7] B. Micusik and T. Pajdla. Structure from Motion with Wide Circular Field of View Cameras. *IEEE Trans. on PAMI*, 28(7):1135–1149, Jul. 2006. 1

[8] R. Mollineda, E. Vidal, and F. Casacuberta. Cyclic Sequence Alignments: Approximate Versus Optimal Techniques. *IJPRAI*, 16(3):291–299, 2002. 3

[9] J. Ng and S. Gong. Learning Intrinsic Video Content Using Levenshtein Distance in Graph Partitioning. In *Proc. of ECCV'02*, volume 4, pages 670–684. Springer-Verlag, 2002. 3

[10] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual Modeling With a Hand-Held Camera. *IJCV*, 59(3):207–232, Sep./Oct. 2004. 1

[11] P. Rousseeuw. Least Median of Squares Regression. *Journal of the American Statistics Association*, 79:871–880, 1984. 5

[12] C. Sagues, A. Murillo, J. Guerrero, T. Goedeme, T. Tuytelaars, and L. V. Gool. Localization with Omnidirectional Images using the Radial Trifocal Tensor. In *Proc. of ICRA'06*, pages 551–556, 2006. 1

[13] F. Schaffalitzky and A. Zisserman. Multi-view Matching for Unordered Image Sets, or "How do I Organize My Holiday Snaps?". In *Proc. of ECCV'02*, volume 1, pages 414–431. Springer-Verlag, 2002. 1

[14] N. Snavely, S. Seitz, and R. Szeliski. Photo Tourism: Exploring Photo Collections in 3D. *ACM Trans. on Graphics*, 25(3):835–846, Aug. 2006. 1

[15] D. Tell and S. Carlsson. Wide Baseline Point Matching Using Affine Invariants Computed from Intensity Profiles. In *Proceedings of ECCV'00*, volume 1, pages 814–828, 2000. 2