

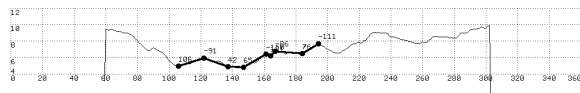


- [1] R C. Arkin, E.M. Riseman, and A. R. Hanson. Aura: And architecture for vision-based robot navigation.. In *Proceedings of the DARPA Image Understanding Workshop*, pages 417-431, Los Altos, 1987.
- [2] R.A. Brooks. A robust layered control system for a mobile robot.. *IEEE Journal of Robotics and Automation*, pages 14- 23, April, 1986.
- [3] T. S. Levitt and D.T. Lawton.. Qualitative navigation for mobile robots. *Artificial Intelligence*, 1990.
- [4] T. S. Levitt, D.T. Lawton, D.M. Chelberg, K V. Koitzsch, and W. D. Dye. Qualitative navigation ii. In *Proceedings of the DARPA Image Understanding Workshop*, pages 319-326, Los Altos, 1988.
- [5] T. S. Levitt, D. T. Lawton, D. M. Chelberg, and P. C. Nelson. Qualitative navigation. In *Proceedings of the DARPA Image Understanding Workshop*, pages 447- 465, Los Altos, 1987.
- [6] H. Nasr, B. Bhanu, and S. Schaffer. Guiding an autonomous land vehicle using knowledge-based landmark recognition.. In *Proceedings of the DARPA Image Understanding Workshop*, pages 432- 439, Los Angeles, 1987.
- [7] F. Stein and G. Medioni. Efficient Two Dimensional Object Recognition.. In *Proceedings of International Conference on Pattern Recognition*, Atlantic City, New Jersey, June 1990.
- [8] F. Stein and G. Medioni. Structural Hashing: Efficient Three Dimensional Object Recognition. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 244- 250, Maui, Hawaii, June 1991.
- [9] F. Stein and G. Medioni. Structural Indexing: Efficient Three Dimensional Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125- 146, February 1992.
- [10] R. Talluri and J. K. Aggarwal. Position estimation for a mobile robot in an unstructured environment. In *IEEE International Workshop on Intelligent Robots and Systems IROS '90*, 1990.
- [11] R. Talluri and J. K. Aggarwal. A positional estimation technique for an autonomous land vehicle in an unstructured environment. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS' 90*, 1990.
- [12] R. Talluri and J. K. Aggarwal. Position estimation for an autonomous mobile robot in an outdoor environment. accepted for publication in *IEEE Journal of Robotics and Automation*, 1992.
- [13] W. B. Thompson, H. L. Pick, B. H. Bennett, M. R. Heinrichs, S. L. Savitt, and K. Smith. Map-based localization: The drop-off problem. In *Proceedings of the DARPA Image Understanding Workshop*, 1990.
- [14] C..Thorpe, S. Shafer, T. Kanade, and M. H. Hebert. Vision and navigation for the carnegie mellon navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1988.
- [15] J. Y. Zheng, M. Barth, and S. Tsuji. Qualitative route scene description using autonomous landmark detection. In *Proceedings of IEEE International Conference on Computer Vision*, December 1990.

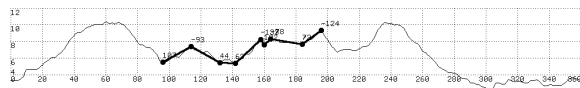
*Location*

*Horizon*

Figure 9 Result



*Matched Super Segment of Observer*



*Corresponding Super Segment of Map*

Figure 10 Hypothesis for Result

Figure 11 Panoramic and Reconstructed View from ALbion Campground

## 6 Conclusion and Future Work

We have presented an algorithm which is able to perform the localization based on the panoramic horizon. We have demonstrated the performance on simulated and real data.

From our experience the algorithm works well. However, the system was not able to perform the correct localization when we put the observer in a small valley where the horizon line is extremely unstable and changes occur caused by minimal alteration of the viewpoint. The general rule is that the further the view is, the more likely it is that the system finds the correct location. To counteract this behavior, an improvement of the sampling of the map could be performed by sampling larger areas with unstable horizons with a smaller grid spacing than areas with a stable horizon. This would improve the performance significantly.

Using significant landmarks in addition to the horizon curves would be a useful extension of the proposed approach. The design of a reliable landmark detector is beyond the scope of this paper

Another issue is the size of the map. Our system fails to recognize the correct horizon, when features (like a high mountain) from outside the map are part of the horizon curve visible by the observer. Therefore we have to use a map which covers a sufficient large area, to include all landmarks, which are visible from the area in which we want to perform the localization task.

Broo86,thorpe88,zheng90,Levitt87,Levitt88,Ark in87

## 7 References

William Thompson from the University of Utah who made the slides and the DEM available.] . The panoramic views were obtained by registering the digitized images manually. For every two adjacent images, we picked corresponding points in the overlapping area, and with a least squares method we computed the transformation.

### 5.2.1 Albion Campground

The panoramic view from Albion Campground is shown in figure [Ref: fig:albionpanorama] (a). It covers more than  $200^\circ$ . The crest line (see figure [Ref: fig:albionpanorama] (b)) was computed by thresholding the image.

The terrain map is the DEM of the surrounding area (see figure [Ref: fig:albiondem] ). The map consists of  $709 \text{ pixel} \times 468 \text{ pixel}$  with a pixel corresponding to a size of  $30 \times 30 \text{ m}^2$  on the ground. The horizons on the map are sampled on a grid with a grid spacing of 10 pixels. This corresponds to 300 m. We compute approximately 1500 horizons. For the linear approximation we use the line fitting tolerances 4, 5, and 6. For the matching we use super segments of cardinality 5, 7 and 9. To obtain useful linear approximations, we scale the relative height of the DEM by a factor of 10.

Due to lack of a camera model we are not able to match the horizon curve directly with the DEM. We have one degree of freedom which is the relative height. We decided to overcome this problem by using several scales for the horizon curve simultaneously. The localization process takes less than one minute and results in the detection shown in figure [Ref: fig:albionresult] . In figure [Ref: fig:albionhypo] we display the detected corresponding super segments of the best hypothesis which cover approximately  $100^\circ$ . Figure [Ref: fig:albboth] shows the comparison between the view of the observer and the rendered view as seen from the computed location.

### 5.2.2 Catherine's Pass

The panoramic view from Catherine's Pass and the corresponding crest line are shown in figure [Ref: fig:cathview] . Our system failed to compute the correct location because part of the horizon corresponds to an area which is not covered by the map (figure [Ref: fig:albiondem] ). This is an example of the limitations of our system.



*Panoramic View*

*Crest Line*

Figure 7 View from Albion Campground

Figure 8 Shaded DEM Map of the Dromedary Peak Area

best match. Until this point we did not consider the case that more than one hypothesis can vote for the same map horizon  $H(v_{best})$ . Using the above mentioned geometrical constraints described above, we check the consistency of the best hypothesis with the rest of the hypotheses in order to find other hypotheses which vote for the same map horizon  $H(v_{best})$ . The location of the map horizon  $v_{best}$  of these consistent hypotheses is the wanted location, where the horizon line resembles most the observed horizon as seen from  $v$ .

This mechanism works well as we show in the next section.

## 5 Results

### 5.1 Results on Simulated Data

The localization algorithm is now illustrated with data examples from a real terrain map. The horizons which are seen by the observer are simulated. For the presentation of the range data we always display the artificially shaded images.

As a terrain map we use the DEM (digital elevation model) covering the Martin Marietta ALV test area. A digital elevation model is a two-dimensional array of uniformly spaced terrain elevation measurements. Our DEM map consists of 810 pixel  $\times$  702 pixel with a pixel corresponding to a size of  $5 \times 5 \text{ m}^2$  on the ground. The whole map corresponds to an area of approximately  $4 \times 3.5 \text{ km}^2$ . Outside of the map we assume flat surface (sometimes the horizon line is not limited to the area described by the map). The horizons on the map are sampled on a grid with a grid spacing of 15 pixels. This corresponds to 75 m. For better visibility we exaggerate the elevation data. The lowest elevation is 21 (in pixels), the highest is 252. The height of the observer above the surface was always constant (2 pixels). For the linear approximation we used the line fitting tolerances 2, 3, 4, and 5. An example is shown in figure [Ref: fig:h1approx] (a) to (d). For the matching we used super segments of cardinality 6 and 8. To obtain useful linear approximations, we scaled the relative height by a factor of 50. These are no critical values, as significant deviations from these values do not affect the results. We choose two examples to illustrate different aspects of our system:

1. Horizon 1 is a general horizon with no specific difficulties.

2. In horizon 2, the observer is located close to a steep ridge.

The shaded map with the super imposed grid of 1211 horizons is shown in figure [Ref: fig:hmap] . The localization in all three examples took less than 1 minute.

#### 5.1.1 Horizon 1:

In the example of horizon 1, the observer is located in a valley between two ridges (see figure [Ref: fig:h1a] ). figure [Ref: fig:h1b] shows the rendered panoramic view from the location of the observer (a) and the extracted horizon curve (b). After the matching the system finds the hypotheses as shown in figure [Ref: fig:h1retrieve] . In figure [Ref: fig:h1c] we display the detected corresponding super segments of the best hypothesis. The overlaid numbers correspond to the angles of the super segments. This hypothesis was found among the hypotheses shown in figure [Ref: fig:h1d] (a) and can be seen in a closeup in figure [Ref: fig:h1d] (b). figure [Ref: fig:h1simulate] shows the rendered view seen by the observer in comparison with the rendered panoramic view of the detected result.

#### 5.1.2 Horizon 2:

Horizon 2 is the horizon seen from the point in figure [Ref: fig:h2a] . The observer is located very close to a ridge. The horizon curve is shown in figure [Ref: fig:h2b] . For this localization example, we do not show all the possible hypotheses, because the correct hypothesis was the only one which “survived” the filter process. A closeup view of the final result is shown in figure [Ref: fig:h2d] . The corresponding rendered views are displayed in figure [Ref: fig:h2simulate] .

#### 5.1.3 Changing the Height of the Observer

We performed some experiments by changing the height of the observer above the surface. Small changes (up to a height of 10 pixels) have no influence on the recognition performance. Increased height (30 and more) adds an increased uncertainty to the localization result. The results show one degree of freedom, and are distributed along a line pointing in the direction of the most stable features (the part of the horizon which is furthest away).

### 5.2 Results on Real Data

In contrast to the last section, we now show two examples which are based on real horizons taken in the Wasatch National Forest, east of Salt Lake City [Footnote: Special thanks to Thomas Colvin and Dr.

2. The super segments are encoded.
3. The encoded super segments are used to retrieve the candidate hypotheses between the super segments of the visible horizon and the super segments of the horizons of the map.

As described in [9], the quantization size is crucial for the encoding of the features. Typical values are (only for the quantization of the angles):

**Table 1:**

cardinality	3	4	5	6	7	8	...
interval size	20	30	40	45	60	60	90

## 4.5 Hypotheses Verification

The task of the verification is to distinguish *good* from *bad* hypotheses. Good hypotheses correspond to true matches, bad hypotheses correspond to wrong matches. To talk about hypotheses we have to define the following terms.

- A hypothesis  $h$  consists of two super segments:  $ss_m$  is the super segment of the map horizon,  $ss$  is the super segment of the observed horizon ( $h = \langle ss_m, ss \rangle$ ).
- The cardinality of a hypothesis  $h$  is defined as:  $c(h) = \text{cardinality}(ss_m) = \text{cardinality}(ss)$ .
- The direction error is defined as:  $\epsilon_d(h) = |\text{dir}(ss_m) - \text{dir}(ss)|$ .
- The angle error is defined as:  $\epsilon_a(h) = (1/c-1) \sum_{i=1}^c |\kappa_i^m - \kappa_i|$  with  $c = \text{cardinality}(ss)$ .
- The vertex error is defined as:  $\epsilon_v(h) = 1/c \sum_{i=1}^c |\Delta \theta_i|$ .

Good hypotheses consist of similar super segments with similar directions and therefore  $\epsilon_d(h) \approx 0$ ,  $\epsilon_a(h) \approx 0$ , and  $\epsilon_v(h) \approx 0$ .

### 4.5.1 Geometrical Constraints:

If more than one hypothesis votes for a specific horizon, we have to verify the consistency of the geometrical constraints. Two hypotheses  $h^1$  and  $h^2$  are consistent when

1.  $|\text{dir}(h^1) - \text{dir}(h^2)| \approx |\text{dir}(ss_m^1) - \text{dir}(ss_m^2)|$  with  $h^i = \langle ss_m^i, ss^i \rangle$

2. and  $|\text{r}(h^1) - \text{r}(h^2)| \approx |\text{r}(ss_m^1) - \text{r}(ss_m^2)|$ .

### 4.5.2 Subsumption:

A super segment  $ss_i$  is subsumed by another super segment  $ss_j$  if the vertices of  $ss_i$  are a subset of the vertices of  $ss_j$ . As an extension we say a hypothesis  $h^1$  is subsumed by another hypothesis  $h^2$  when their super segments are subsumed by each other:  $h^1$  is subsumed by  $h^2$  if  $ss_i^1$  is subsumed by  $ss_j^2$ , and  $ss_j^1$  is subsumed by  $ss_i^2$ . When one hypothesis is subsumed by another hypothesis we remove the hypothesis with the smaller cardinality.

Figure 6 Hypothesis Verification

Therefore, the verification stage consists of the following steps (see Figure 6).

1. We compute all possible matches for the observed horizon with the map horizons to generate the hypotheses.
2. We filter out the hypotheses which are subsumed by other hypotheses. We also remove hypotheses which have large errors for  $\epsilon_d$ ,  $\epsilon_a$ , or  $\epsilon_v$  (typically we require that  $\epsilon_d, \epsilon_a, \epsilon_v < 20^\circ$ ).
3. After removing the hypotheses which are unlikely to represent a good match, we want to find the best hypothesis of the remaining ones. This is done using a simple heuristic. We can sort the hypotheses based on the values of  $\epsilon_d$ ,  $\epsilon_a$ ,  $\epsilon_v$ , and the cardinality. We are interested in the largest possible match (maximal cardinality) with the minimal error (minimal  $\epsilon_d, \epsilon_a, \epsilon_v$ ).

$$w(h) = w_d \epsilon_d(h) + w_a \epsilon_a(h) + w_v \epsilon_v(h) + w_c c(h)$$

Typically we use  $w_d = w_a = w_v = -1$  and  $w_c = 2$  to punish errors and to reward high cardinality. This results in an ordered set of hypotheses. We assume that the best hypothesis represents the

### 4.2.5 Angles

Let a super segment consists of  $n$  segments, then we have  $n-1$  angles between successive segments.

### 4.2.6 Direction

A super segment represents a part of a horizon. The first vertex is located at the direction  $\theta_{start}$ , the last vertex is located at  $\theta_{end}$ . We define the direction of a super segment  $ss$  as the middle between the start and the end vertex:

$$\text{dir}(ss) = \begin{cases} \frac{\theta_{start} + \theta_{end}}{2} & \text{if } \theta_{start} < \theta_{end} \\ \frac{(\theta_{start} + \theta_{end} + 360) \bmod 360}{2} & \text{otherwise} \end{cases}$$

### 4.2.7 Relative Height Range

The relative height  $r(ss)$  of a super segment  $ss$  is defined as the middle between the minimal and the maximal relative height.

$$r(ss) = \frac{r_{\max}(ss) + r_{\min}(ss)}{2}$$

The relative height range  $\Delta r(ss)$  is defined as the difference between the maximum relative height and the minimum relative height:

$$\Delta r(ss) = r_{\max}(ss) - r_{\min}(ss)$$

As mentioned before, we are mainly interested in the curvature information implicitly captured by the super segment angles. This is the reason why we use them to encode a super segment. To avoid establishing matches between super segments which have the same angles but totally different ranges of relative height, we add the relative height  $\Delta r$  to our coding scheme. To encode a super segment  $ss$  with cardinality  $n$  we use a simple encoding scheme. The list of the quantized curvature angles and the relative height range is the code of the super segment  $ss$ :

$$\text{Code}(ss) = (\text{Quant}(\kappa_1), \text{Quant}(\kappa_2), \dots, \text{Quant}(\kappa_{n-1}), \text{Quant}(\Delta r))$$

All the encoded super segments serve as keys into a table (the data base), where we record the corresponding super segments as entries, as explained later.

## 4.3 Representation of the Map

We represent the map by a set of horizons computed for the points of a grid superimposed on the map. The spacing of the grid determines the accuracy with which we can find the correct location. To find the correct horizon which corresponds to the horizon observed by the observer, we want the encoding to be compact, accessible fast, and the stor-

age of a large number of horizons should be possible. We choose for these reasons a data structure which is implemented as a hash table. A hash table allows efficient storage (only pointers are recorded), the hashing scheme allows fast access and different super segments with the same keys can be stored in cellar like buckets. The encoding of a map consists of the following steps (see Figure 5):

1. Compute the horizons  $H_k$  of the map.
2. Compute the super segments  $^{k}ss_i$  of  $H_k$ .
3. Encode the super segments. For every super segment several codes  $Code_j$  are computed. They are related to the different line fitting tolerances. Every code of every super segment serves as a key for an entry in a table (the data base) where we record the the super segment and the corresponding horizon.

The table (data base) grows in size with the number of recorded horizons. This process of building the data base from the map is performed off line.

Figure 5 Map Encoding

## 4.4 Hypotheses Generation

The task of hypotheses generation is the process to establish correspondences between horizons of stored horizons and the horizon seen by the observer. This process results in a set of matching hypotheses which consist of *good* and *false* matches. To extract the good hypotheses from all hypotheses we have to filter out the good hypotheses based on geometrical and similarity constraints. This process is discussed in section 4.5. Our main interest for the candidate retrieval lies in the discriminative power of the hypotheses generation itself. By using indexing, we gain a lot of this power. We believe that our indexing mechanism reduces the ratio of false hypotheses to good hypotheses tremendously. The hypotheses generation consists of the following steps:

1. The horizon is preprocessed to generate all the super segments as explained in section 4.2.

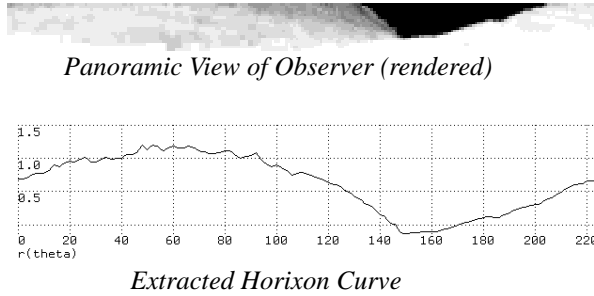


Figure 4 Graph of Horizon 1

## 4 The Algorithm

### 4.1 The Basic Idea

As mentioned in [13] there is an important relationship between localization and recognition. Many of the computational tools that have proven useful for recognition turn out to be also relevant to localization. In outdoor navigation, the relevant “model” is a representation of the topographic features visible from a particular vantage point. The number of vantage points is effectively unbounded. So far however, the common belief is that this fact does not allow the use of a bounded set of models for the localization problem. This is why most approaches try to assemble the topographical information adaptively from the map. We show in this paper that the localization problem can be reduced to a recognition problem when the system is able to deal with a data base which includes a large set of horizon models. Our localization approach is related to our early work [7], which addresses the problem of recognition of multiple flat objects in a cluttered environment from an arbitrary viewpoint. It allows to perform fast recognition even with large data bases. We exploit this fact to store a large set of horizons from a grid of vantage points and by using indexing we find the “most similar” horizon.

### 4.2 Representation of a Horizon

#### 4.2.1 Offline Computation from the Map:

A horizon is computed from a map by computing a set of sample slices at equal angles around the location  $v$ . Starting at  $0^\circ$  (north) a sample slice is computed through the landscape (see Figure 2) every  $\Delta\theta$  angle (typically  $0.5^\circ \leq \Delta\theta \leq 3^\circ$ ). For every slice at the

angle  $\theta$  the point  $p^\theta$  is obtained. From  $p^\theta$  and  $v$  we compute the value for the tangent  $r^\theta$ . The set of all points  $(\theta, r^\theta)$  with  $0^\circ \leq \theta < 360^\circ$  represents the horizon.

#### 4.2.2 Computation of the Observable Horizon:

An observer located in an unknown environment can extract the panoramic horizon by

1. acquiring a panoramic image,
2. segmenting the “sky” from the “ground”,
3. linking the horizon curve,
4. and sampling the horizon curve.

In this paper, we are not addressing the crucial segmentation step.

#### 4.2.3 Encoding:

Our encoding of a horizon is based on a polygonal approximation. We are not dependent on any feature detection algorithm and we do not have to handle explicit distinguished points like corners or inflection points. Our opinion is that curvature is the most important feature of a general curve. It is invariant with regard to scale, rotation and translation. By using a polygonal approximation we lose most of the curvature information, but we keep parts of this information in the angles of consecutive line segments.

Obviously, there is not a unique polygonal approximation for a horizon curve. Therefore, for the purpose of robustness, we use *several* polygonal approximations with different line fitting tolerances. Since we want to handle occlusion, we do not expect to obtain complete horizons, but only portions of them. On the other hand, individual segments are too local to be useful as matching primitives. Grouping a fixed number of adjacent segments provides us with our basic features, the super segments. In order to manipulate and to encode super segments we have to define some terms and to describe some attributes.

#### 4.2.4 Cardinality

The cardinality of a super segment is the number of segments it consists of. This cardinality embodies the trade-off between the system’s ability to deal with occlusion and its discriminative power: long super segments are very discriminative and allow rapid recognition, but they also cover a large portion of the horizon and may therefore be occluded. Short super segments, on the other hand, are less distinctive, but also less likely to become occluded.

to this patch.

**Viewer or Observer:** The viewer is an observing platform with a camera. In addition it can retrieve direction information based on a compass (this ability speeds up the localization significantly). The observer has the coordinates  $v=(x_v, y_v, z_v)$ . The camera is located above the surface:  $z_v = \text{Map}(x_v, y_v) + h$ , where  $h$  is the height of the camera.

Figure 1 Horizon

Figure 2 Slice through a Landscape at an angle  $\theta$

**Horizon:** The horizon is the upper bound of the projection of all landscape points on a cylinder around the observer as shown in Figure 1. Such a panoramic horizon  $H$  is a periodic curve. Projecting the curve back on the landscape results in a set of three dimensional points  $\bar{H}$  whose projection on the map is shown in the results section.  $H$  and  $\bar{H}$  are defined in the following way (see Figure 2):

$$H(v) = \{ \Delta z(d)/d \mid \max_d \{ \Delta z(d)/d \}, \\ 0^\circ \leq \theta < 360^\circ \} \\ \bar{H}(v) = \{ p_v(\theta, d) \mid \max_d \{ \Delta z(d)/d \}, \\ 0^\circ \leq \theta < 360^\circ \}$$

with

$$v = (x_v, y_v, z_v) \\ p_v(\theta, d) = (x_v \cos \theta + y_v \sin \theta, z_v) \\ d = \sqrt{(x_v \cos \theta - x_v)^2 + (y_v \sin \theta - y_v)^2} \\ \Delta z(d) = z_v - z_v \theta$$

We call the tangent of  $\alpha$  the *relative height*  $r$ , with  $r(d) = \Delta z(d) / d$ . Without loss of generality we set  $0^\circ = 360^\circ$  as the north direction and define the orientation of a horizon as clockwise (as seen from above). An example can be seen in Figure 3. We display a horizon as a graph as seen from the observer. The abscissa represents the direction  $\theta$  ( $0^\circ = \text{north}$ ), the ordinate represents the relative height  $r$ . The graph of the horizon of Figure 3 can be seen in Figure 4.

It should be noted that a horizon can be occluded and changed by environmental influences such as fog or clouds. However, our algorithm does not rely on complete data. As long as significant parts of the horizon are visible, the correct localization can be performed.

In practice, problems may occur when the camera platform is not correctly levelled, as large inclination angles significantly alter the horizon curves. We therefore assume a small leveling error, and the resulting errors are absorbed by our encoding scheme.

Figure 3 Map with Superimposed Reprojected Horizon 1

server at each location. Such curves are encoded and stored in a table. To locate an unknown location, we first extract the panoramic horizon curve of the unknown location. Then we approximate it by a family of polygons with different line fitting tolerances. By using indexing into the table, we retrieve candidate locations. The correct candidate (the closest one) is found by applying further geometrical constraints in the verification step.

It should be clear to the reader that this approach is not guaranteed to provide a unique answer, as it is easy to come up with counter examples (planar or repetitive environment), but we show that it gives excellent results in complex environments. We start by reviewing some of the relevant work in this area, then precisely define the terms we will be using, and clearly state the problem. We then proceed to explain our approach by discussing the basic idea, and present the algorithm. We validate our claims by showing some results from a real map.

## 2 Related Work

Given a map and a certain view from the viewpoint of an observer, the question is: what is the exact position of the observer? Most of these systems assume a known initial position and try to update their location while they move in order to perform correct navigation.

Thorpe *et al* [14] present results on the visual navigation for mobile robots in outside environments. They discuss two algorithms: color vision for road following, and 3D vision for obstacle detection and avoidance. The resulting system is able to navigate continuously on roads while avoiding obstacles.

Zheng *et al* [15] introduce an approach to build qualitative descriptions of scenes along a route, which is used in route recognition by a mobile robot. The robot uses autonomously selected landmarks from the panoramic views for navigation. The landmarks are detected by using an algorithm which detects “unique patterns” in the images.

Arkin *et al* [1] explore several visual strategies for the navigation of a mobile robot. These include a line-finding algorithm for path following, a multiple frame depth-from-motion algorithm for obstacle avoidance, and the relationship of schema-based scene interpretation to mobile robotics, especially regarding vehicle localization and landmark recognition. They present results of actual robot navigation in an outdoor environment.

Nasr *et al* [6] present a landmark recognition technique based on a perception-reasoning-action and expectation paradigm of an intelligent agent. It

uses extensive map and domain dependent knowledge in a model-based approach. They present examples using real ALV (Autonomous Land Vehicle) images.

Levitt *et al* [3] developed a theory of representation of large-scale space based upon the observation and re-acquisition of distinctive visual events, i.e. landmarks. They are able to integrate the visual information from many images into a uniform database as a sensor-based robot moves through its environment. The representation provides the foundations for visual memory databases and path planning. They demonstrate their claims with a navigation simulator.

Talluri and Aggarwal [10] [11] [12] recently developed a system which is the closest to our approach in solving the localization problem. They take different views of the horizon and use them to search the underlying map for possible robot locations. In general several of these views constrain the possible locations to a small area. They show some results on simulated data based on a digital elevation map.

Thompson *et al* presents a formalism in [13] within which the localization (“drop off problem”) can be studied. They discuss the approach which is taken by an expert human map users to deal with localization, and they propose a preliminary computational model of the process.

Reviewing the existing systems, most work was done regarding the localization based on specific landmarks. The focus of our research is to solve the localization problem using the panoramic horizon, given a topographic map and the orientation of the observer.

## 3 Glossary

Let us first define the following terms:

**Localization:** Localization is the process of establishing a match between a particular location in the environment and the corresponding location on a map [13].

**Map:** A map in our definition is a topographic map which provides us with the three dimensional coordinates for each surface point (e.g. range data). The  $z$  component (altitude or height) can be retrieved from the  $x$  and  $y$  coordinates:  $z = \text{Map}(x, y)$ . We assume that the map is small with respect to the associated planet sphere. Therefore the map can be considered as a small planar patch. The  $z$  axis is perpendicular

# Map-Based Localization using the Panoramic Horizon

Fridtjof Stein and Gérard Medioni  
Institute for Robotics and Intelligent Systems  
Powell Hall 204  
University of Southern California  
Los Angeles, California 90089-0273  
Email: stein@iris.usc.edu

**Abstract:** We present an approach to solve the localization problem, in which an observer is given a topographic map of an area and dropped off at an unknown location. The solution to this problem requires establishing correspondences between viewer-centered observable features and their location on the map. The feature we select is the panoramic horizon curve, defined as the sky-ground boundary perceived by the observer as he performs a full 360° in place. In our approach, we first precompute, offline, these horizon curves at a set of locations on a grid, from the topological map. These curves are approximated by polygons with different line fitting tolerances to gain robustness to noise in our representation. These polygons are grouped into overlapping super segments, which are then encoded and stored in a table. The online computation consists of acquiring the panoramic view and extracting (with human help) the horizon curve. This curve is approximated by a polygon and the resulting super segments, used as indices in the data base, allow us to retrieve candidate locations. The best candidate is selected during a verification step which applies geometric constraints. This process is using local features and can therefore tolerate significant occlusion likely to occur in real environments. We illustrate the performance of the approach on results obtained from real data.

## 1 Introduction

In this paper we look at a specific application of structural indexing, namely the localization of an observer in an unfamiliar environment. We demonstrate how an original 3D problem can be reduced to two dimensions and solved with a system similar to the one presented in [7].

Navigation using maps requires the frequent updating of the location of an observer with respect to the map. Humans performing this task use a large number of different problem solving techniques, both expectation driven, where a landmark is selected on the map and searched for, or data driven, where image features are extracted first and matched against the map.

The problem most often addressed by previous researchers is that of updating the current position, given a good initial solution. This can be solved by correcting (small) errors between the predicted and observed aspects of the scene.

Here instead, we study the “drop-off” problem. An excellent overview of the whole problem is given in [13]. The authors describe a preliminary computational model for the drop-off problem (the name *drop-off* comes from the case in which an observer is “dropped off” into a unfamiliar environment and has to orient itself). The observer stays at one position and tries to find its location based on visible landmarks and salient features. In our approach, the observer tries to establish its location based on the curve described by the panoramic horizon (as explained later) which is visible from his viewpoint.

It should be noted that people experience serious difficulties in solving such localization problems, leading many researchers to suggest that traditional object recognition strategies are unlikely to succeed [13]. This is based on the observation that the combinatorics of the problem are very unfavorable, as the shapes are complex and the number of different aspects is extremely large.

In this paper, we challenge this view and propose that a table-based matching strategy [7][8] can overcome the limitations mentioned previously. We propose to extract from many locations in the map the panoramic horizon curves, the crest line perceived by the observer as he completes a full 360° view in place, which would be observed by the ob-

---

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Contract No. F49620-90-C-0078 and by a NSF Grant under award No. IRI-9024369. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. We gratefully acknowledge grant No. 89-00481/1 from the US-Israel Binational Science Foundation (BSF), Jerusalem, Israel.