ACKNOWLEDGMENT

The authors would like to thank the referees for their useful comments as well as the members of the Computer Vision Laboratory for their discussions during the development of this work.

REFERENCES


Abstract—This paper discusses the use of computer graphics in estimating the position of an autonomous mobile robot navigating in an outdoor, mountainous environment, which is fundamentally a computer vision task. A digital elevation map (DEM) of the area in which the robot is to navigate is given, and the robot is equipped with a camera that can be panned and tilted, a compass, and an altimeter. The position of the robot is estimated by establishing a correspondence between the images acquired by the camera on the robot (actual images) and the images generated from the DEM (predicted images) using computer graphics techniques. Features are extracted from the predicted images, and the actual images that are used in establishing the correspondence. The features used are the horizon line contours (HLC's) in the images. In order to reduce the search space (the set of possible robot locations) to be considered in generating the images, a constrained search paradigm is used. Geometric constraints help prune the search space significantly. The novel feature of this work is the collaboration between computer graphics and computer vision to establish the position of the robot in its environment. The approach is tested using real terrain data of areas in Colorado and simulated images. The method is suitable for use in outdoor mobile robots and planetary rovers.

Index Terms—Autonomous navigation, computer graphics, computer vision, constrained search, digital elevation map, position estimation.
cameras to those obtained by rendering the 3-D scene using graphics. Using this comparison, he modifies the parameters describing the object scene to reduce the error between the natural and synthetic scenes.

For mobile robot self-location, Kak et al. [2] and Tsubouchi and Yuta [3] have made use of computer graphics in the solution to the image/map correspondence problem. Both of these papers deal with indoor mobile robots with a given CAD model of the environment. Kak et al. [2] present PSEIKI, which is a system that uses evidential reasoning in a hierarchical framework for image interpretation, and discuss how it may be used for self-location by a mobile robot. The robot's position encoders are used to maintain an approximate estimate of its position and heading at each point. A visual sensor, in conjunction with a CAD model of the building, is used to derive a more accurate estimate of the robot's position and pose. The approximate position from the encoders is utilized to generate an estimated visual scene from the CAD model, which is then matched against the actual scene viewed by the camera. Once the matches are established between the features of the two images (expected and actual), the position of the robot may be estimated with a reduced uncertainty. Tsubouchi and Yuta [3] discuss a similar position estimation technique used in their YAMABICO robot. This technique matches trapezoids extracted from the actual color images with those generated from the map using computer graphics. The common idea among all three investigations is the matching of a synthesized image with the observed image.

This correspondence considers the general problem of mobile robot self-location by image/map correspondence, with emphasis on the use of computer graphics techniques. The problem we address is that of position estimation of an autonomous land vehicle navigating in an outdoor mountainous environment. A digital elevation map (DEM) of the area in which the robot is to navigate is given. The DEM consists of elevation data at a discrete set of points gridded along the North-South and East-West directions. Fig. 1 shows a synthetic top view of the DEM used in our experiments, which was rendered using Gouraud shaded polygons. The robot is equipped with a camera that can be panned and tilted, a device to measure the robot's altitude, such as an altimeter, and a compass. No recognizable landmarks are assumed to be present in the navigation environment. Computer graphics techniques are used to render the 3-D information in the DEM and generate predicted images, which are then matched with the actual images acquired by the robot to estimate its position. We extract features (horizon line contours (HLC's)) from the predicted images and the actual image and establish a correspondence between these features.

Since an exhaustive search of the entire map can be prohibitively expensive if the map is large, we use a two-stage constrained search strategy. The first stage is a constrained search of the DEM for a reduced set of possible robot locations, that is, we extract features from the images taken by the robot (HLC's), and instead of searching for the corresponding features in the map, we search the space of the possible robot positions in the DEM for the locations from where such features could be imaged. Rather than using the entire feature, a subset of the feature (i.e., the height of the DEM in the center of the image plane) and the known camera geometry of the perspective projection are used as search parameters. We derive geometric constraints from the image feature and the camera geometry that need to be satisfied by any hypothesized position of the robot. Using these constraints, we eliminate large subspaces of the search space of possible locations.

In the second stage, we use a hypothesize-and-verify strategy to disambiguate among the reduced set of possible robot locations returned by the first stage. Using computer graphics, we generate the predicted image for each of the possible robot locations and compare its HLC with the HLC of the actual image. The location that yields a predicted image that most closely matches to the actual image is then considered to be the best estimate of the robot's location.

This technique of exploiting the collaboration between computer graphics and computer vision to isolate the position of the robot in its environment is quite novel and has many advantages over the previous techniques used for position estimation. It does not rely on the existence and detection of environmental landmarks. It does not necessitate the building of a complicated world model from the sensor observations. It is a totally passive navigation technique. Any errors in estimating the position are not cumulative. It also does not require an a priori estimate of the position, as do the previous techniques reviewed above. However, if such an estimate is available, it can be used to speed up the search process as will be discussed in Section II. The initial results of this approach were presented in [4]-[6].

II. POSITION ESTIMATION USING HORIZON LINE CONTOUR

In the first stage of the search, images are acquired in the four geographic directions (N, S, E, and W) of the current robot position. In acquiring these four geographic views, the camera is assumed to have zero roll, and the optical axis of the camera is assumed to pass through the image center. The camera's tilt angle \( \phi \), which is defined as the angle between the optical axis and the horizontal plane, is adjusted until the horizon line is clearly visible in the image. An inclinometer may be used to measure the tilt and roll angles. Commercial inclinometers with an accuracy of up to 0.005° are available. The altitude of the camera is assumed to be measured using an altimeter. The HLC is then extracted from these images using a gradient operator, and the height of the HLC in each of the four images is measured. Let these heights be \( h_i (i = N, S, E, W) \). The reason for using the height of the HLC at the center of the image plane is that the DEM is assumed to be gridded along N-S and E-W axes. Therefore, the points that project onto the HLC at the center all lie along the same grid line in the DEM. Using the altitude of the camera \( H \), the tilt angle \( \phi \), and the HLC height \( h_i \), one is of the directions, e.g., North, the DEM is searched for possible camera locations, that is, a camera location is hypothesized at each of the DEM grid points, and the height of the HLC \( h_i \) is back projected onto the DEM using the camera geometry to see if any hypothesized points can project to this height \( Z_{opt} \). If any such points exist, the camera location is marked as a possible candidate. Equation (1) gives the estimated height \( Z_{opt} \) for a distance \( x \) from the hypothesized camera location. Details of the derivation of the derivation are given in [6].

\[
Z_{opt} = H + x \sin \phi + x \frac{\tan \theta}{\cos \phi} \frac{h}{I_{max}} - \sqrt{\frac{h^2 + (f \phi)^2}{h^2 + f^2}}
\]

where \( I_{max} = \text{image Plane Height} \), \( f \) is the focal length, \( \theta \) is the perspective angle, \( h \) is the HLC height at the center of the image.
plane, \( \phi \) is the camera tilt angle, \( H \) is the altitude of the camera above the datum, and \( x \) is the distance between the point and the camera.

At any grid point, if the height is larger than that estimated by the back projection, then all the camera locations between the current location and this tall point are marked as impossible positions and discarded. This process is repeated for all the locations in the map that are not discarded. This constraint reduces the search space significantly. Another constraint used to reduce the search space is to use the altitude \( H \) of the camera, as measured by the altimeter. The possible robot locations are thus restricted to those points in the DEM where the actual elevation \( Z_{\text{actual}} \) lies close to \( H \).

The results of the search process using one of the images are thus a sparser set of possible camera locations. These are then considered as the possible set for the next search, which searches this set using the geometric constraints extracted from the image in another direction. This process is continued by successively applying the constraints from all four directions, which refines the possible locations to a small set that is usually clustered around the actual location. In our experiments, we found that the search process is not sensitive to the order of the directions in which the search is performed. Note that the search is performed in only four directions since the elevation map is a rectangular grid, and we wish to search only along the grid lines in this stage of the search.

The search algorithm is sensitive to the errors in three measurements: the height \( H \) of the camera, the HLC height \( h \), in the image plane, and the tilt angle of the camera \( \phi \). These three parameters affect the estimated elevation \( Z_{\text{est}} \) at a particular point in the DEM for a given hypothesized camera position. The errors in \( Z_{\text{est}} \) directly reflect as errors in the position estimation since only when \( Z_{\text{est}} \) is equal to \( Z_{\text{actual}} \) do we consider the current camera location as a possible candidate. The error in \( H \) is directly reflected as an error in \( Z_{\text{est}} \). Therefore, to account for these, a worst-case error in \( H \) is estimated from the altimeter. Let this be \( \beta_H \). In the search algorithm, this can be accounted for by considering the estimated elevation at the candidate point to be acceptable if \( Z_{\text{est}} - \beta_H < Z_{\text{actual}} < Z_{\text{est}} + \beta_H \). The errors in \( h \), are mainly due to image quantization error and errors in the edge detector used to extract the horizon line. Since \( x \) (the distance between the DEM point and the camera) occurs as a multiplicative parameter in the second and third terms of (1), the errors in \( h \), are magnified by this and, hence, vary depending on \( x \). The worst error compensation should take this into account. One way to account for these errors in \( h \), is to back project a band of \( h \), values \( (h_{-1}, h_0, h_1 + \Delta h) \) instead of a single \( h \), in the estimation of \( Z_{\text{est}} \). Thus, for each camera and DEM point location, we obtain a range of acceptable elevations \( (Z_{\text{est}} + \Delta Z_{\text{est}}, Z_{\text{est}} - \Delta Z_{\text{est}}) \), and we consider the current camera location as being possible only if \( Z_{\text{actual}} \) at this DEM point lies within this range. In such a manner, we also implicitly account for the effect of \( x \) on \( \Delta Z_{\text{est}} \). Similar to \( h \), the effects of errors in \( \phi \), are also magnified by \( x \). The errors in \( \phi \), are accounted for by considering the effect of these errors as errors in \( h \), that is, the band \( h \_i \pm \Delta h \) is made wider to account for \( \Delta \phi \). Hence, the range of acceptable \( Z_{\text{est}} \) values is increased. Note that by back projecting a band of values, we increase the set of possible robot locations returned after the search in each direction.

In the second stage, each point in the set of possible locations is considered to be a likely candidate, and the image that would be seen in a particular direction if the camera were located at that location is generated from the DEM using computer graphics rendering techniques. The HLC's from these images are then extracted and compared with the actual image HLC in the same direction, using a curve matching technique.

![Fig. 2. Typical view and the extracted HLC.](image)

We use the special-purpose graphics hardware and software of the AT&T Pixel machine to render the image from the elevation data. In generating the images from the DEM, the elevation data is tessellated into polygons, and surface normals are calculated at each of the elevation points using the four neighboring points. A light source position and direction are assumed, and for a given camera location and perspective geometry, Gouraud-shaded polygons are drawn and projected onto the image plane to generate perspective views of the mountain range. The HLC is extracted from these images using a gradient operator.

Matching the HLC's extracted from the predicted image and the actual image is basically a 2-D curve-matching problem between the HLC extracted from the image (model curve) and the HLC's generated from the candidate locations using the DEM (candidate curves). The objective is to find the candidate curve that most closely matches the model curve. There is a substantial body of work on matching curves for object recognition [7]. It mainly concerns matching an image curve to a model and estimating the position and pose of the camera for the best match. Our problem differs from these problems in that we use a search strategy to first isolate similar curves. Our objective is only to isolate the curve that best matches the model curve. We use a least square technique to determine the best match. Essentially, we compute the mean square disparity between each of the candidate curves and the model curve. The candidate curve that results in the lowest mean square error is considered to be the best estimate of the robot's position.

To reduce the effects of noise on the HLC, the HLC's are first smoothed using a Gaussian low-pass filter. Due to quantization and other noise effects, this search strategy does not always isolate the position to the grid point nearest to the exact location, but the returned position is still within a small neighborhood around the exact location. It is possible to apply the stage-2 search to the neighborhood of the best estimate and to further refine the estimate and get a lower error.

A. Illustrations

The algorithms developed in this research have been tested using real terrain data obtained from the United States Geological Survey (USGS) of various areas in Colorado and Texas and simulated images. This section details the results of a typical run using a DEM of an area in Colorado. The elevation data is a uniform square grid of 30 m resolution and has 359 \( \times \) 457 grid points. It covers an area of 148 km\(^2\). Synthetic images for an assumed camera location are generated from the DEM using the AT&T Pixel machine to serve as the images taken by the robot. Figs 2-7 illustrate a typical run of the algorithm. Fig. 2 shows a typical image used to test the algorithm and the HLC extracted from this image using a gradient operator.

In this example of the 164,063 (359 \( \times \) 457) possible locations, the first stage of the search process, using \( h \_i \) and the associated camera geometry, returns 1104 possible camera locations. Fig. 3 shows a top
Fig. 3. Possible camera locations after searching in N (top view).

Fig. 4. Possible camera locations after searching in N and E (top view).

Fig. 5. Possible camera locations after searching in N, E, and S (top view).

Fig. 6. Possible camera locations after searching in N, E, S, and W (top view).

Fig. 7. Disparity between HLC’s.

The estimate of the location is isolated as the one with the least mean square error.

The algorithm was also tested by adding zero mean Gaussian noise with a standard deviation of 5 pixels to the horizon lines. This represents the noise in the image formation process and the noise in the detection of the HLC. Since the HLC’s are smoothed by a Gaussian filter and since we actually use a band of \( h \) values in the back projection, the effects of the additive Gaussian noise are not significant. It was found that the effect of this noise is to slightly shift the estimated position in some cases \[6\]. However, the estimated position is still quite close to the actual location in all cases.

This search strategy can be also used in a bootstrap mode, that is, once the robot isolates its position in the DEM, in the subsequent navigation tasks this position estimate can be used to search only near the current robot location rather than the entire DEM. When the location of the robot is known quite accurately, we can, in fact, eliminate the first state and only the use second stage of the search in a small neighborhood around the current location. Sometimes, the search process may return more than one location as a possible robot location. One way to disambiguate between these possible locations is to translate the robot by a known amount in a known direction, image the environment, and then reapply the search strategy with additional constraints imposed by the new measurements.

III. CONCLUSIONS

In this paper, we presented a novel method using computer graphics to assist in the computer vision task of estimating the position of a mobile robot in an outdoor mountainous environment. The robot’s position is estimated by establishing correspondence between the DEM and the images acquired by the camera. Computer graphics techniques are used to transform the DEM information into the image domain. The height of the center and the shape of the horizon line contour (HLC) in the image plane are used as the main features in establishing this correspondence. A constrained search paradigm is used to reduce the search space of the possible robot locations.
Although we used the AT&T Pixel machine’s hardware and software to render the DEM and then extracted the HLC from the rendered images, we can also generate the HLC alone without rendering the entire image. This is, in fact, very similar to the problem of plotting a single-valued, continuous function of two variables under perspective projection [8]–[11]. Our problem actually constitutes a special case of the hidden surface removal problem in computer graphics; this is a problem for which very fast solutions are possible. However, these fast solutions are usually image-precision algorithms, and the limited resolution of the data structures used makes the solutions prone to false line visibility and aliasing problems [11], which reduces the rendered HLC’s accuracy. By using the AT&T Pixel machine to render the image, our approach automatically takes care of the visibility problems by the hardware :buffering algorithm. In addition, the Pixel machine uses object-precision in all its computations, thus reducing the aliasing effects. Antialiasing by super sampling also further reduces the aliasing effects and results in a more accurate rendering of the image. The gradient operator used to extract the HLC also gives subpixel resolution in the HLC.

The position estimation can be made more robust by using features other than the HLC alone. One approach is to detect curves from other parts of the image that arise from intensity changes in the image and to search for the DEM depth discontinuities that cause these intensity changes. Another approach is to generate a stereo pair of synthetic images from the DEM, to establish correspondence between them, and to recover a sparse depth map from this correspondence. This recovered map can then be matched to the DEM to get a more accurate position estimation. Gennery [12] discusses such a system using real images from a Mars rover. Rodriguez and Aggarwal [13] describe such a system used in aerial navigation.

Our technique implicitly assumes that the DEM’s sampling rate does not significantly change the HLC’s shape between grid points. If this is false, both stages of the search process will be affected. In stage one, since we search using the DEM heights along a grid line to isolate the possible robot locations, if the robot is situated between two grid points and if the sampling rate is too small, this search process may not isolate the true position. In stage two, to isolate the true robot location, recall that we compare the candidate and the actual HLC’s using a curve matching technique. In generating the candidate HLC, we use Gouraud shaded polygons to render the DEM and extract the HLC from it. This is equivalent to linear interpolation between the DEM points. Thus, if the DEM sampling rate is too small, the disparity between the candidate and the true HLC may be large and may not be a good representation of the robot’s true position. One way to alleviate this problem is to resample the existing DEM to a higher resolution by using an interpolation scheme more representative than the linear interpolation accomplished by the rendering process. Yokoya [14] suggests using a fractal interpolation scheme to generate a higher resolution DEM from an existing coarse-resolution DEM. The newly generated DEM is statistically more like the original DEM and thus may work better in our approach.

However, note that the problem of a low DEM sampling rate is similar to the problem of a quasi-uniform and approximate tessellation of the viewing space in aspect graph-based approaches to object recognition [15]. In these approaches, researchers use an approximate and quasi-uniform sampling of the viewing space to represent the object’s possible aspects since forming an exact aspect representation can be complicated and the number of possible aspects can become very large. Since the object’s different aspects may change significantly in size and shape, any uniform sampling may miss some of these aspects. Thus, the object recognition strategies that use these techniques are error prone. However, because natural terrain usually changes gradually, the HLC’s shape may not change significantly between DEM samples. In addition, since natural terrain usually exhibits certain statistical self-similarity, the fractal-based interpolation scheme suggested above [14] may help to alleviate the problem. Since the method relies on the horizon line’s uniqueness, the approach is limited to areas of reasonable altitude variations.

REFERENCES