Estimating Camera Position and Orientation from Geographical Map and Mountain Image*

Prospero C. NAVAL Jr., Masayuki MUKUNOKI, Michihiko MINOH, Katsuo IKEDA
Department of Information Science, Kyoto University

Abstract: We describe a method for recovering the camera position and orientation parameters from a single mountain image. It is based on the alignment of the mountain skyline with a synthetic skyline generated from a digital elevation map. Image plane alignment of three image skyline feature points with three model feature points is first hypothesized and the position and orientation parameters for the hypothesis are computed by nonlinear least squares optimization. The hypothesis is then verified by checking for skyline alignment. Search space is reduced using a strategy presented in this paper. Simple skyline and feature point extraction procedures are also discussed.

1 Introduction

This paper considers the problem of accurately estimating the position and orientation of a camera from an image of a mountain scene given the digital elevation map (DEM) of its surroundings. Solving this problem is important for certain vision-based positioning tasks and other applications. In tank navigation, for example, GPS signals may be unavailable during battle conditions and inertial position sensors may be unreliable since they drift with time (1 nautical mile per hour or more of drift) [1]. It is also useful when one wants to know the location where an image without positional information (GPS,landmarks, etc.) was taken. Previous researches that addressed the position estimation problem often require a good initial estimate of the position or elevation ([2], [3], [4], etc). These approaches are restricted in applicability since this requirement may not be met. One approach that does not have this limitation is the table-based matching strategy of Nevatia et. al. [5]. In their method, the 360° panoramic model skylines at many locations are generated from the DEM and stored in a table. To locate an unknown position, the panoramic skyline for that position is extracted from multiple images using the camera and used as an index to the location. The main disadvantage of this technique is that a very large database is needed to cover a large region.

We propose a method for camera position estimation based on the matching of image skyline with a synthetic skyline generated from the DEM. It does not require storing synthetic skylines in a database nor the extraction of the location's panoramic skyline (only one image is necessary). Additional a priori knowledge (e.g. initial estimate of position or that a certain mountain is visible in the image, etc) can be exploited to reduce processing time and/or improve the accuracy of the final output. It can also tolerate partial occlusion.

Our technique follows the object recognition by alignment scheme of model-based vision. First, image plane alignment of three image feature points with three model (i.e. DEM) feature points is hypothesized. The pose for this hypothesis is then computed using nonlinear least squares optimization and verified by checking if the model skyline that the pose produces aligns with the actual skyline. We adopt a k-nearest feature point search strategy to avoid combinatorial explosion. The feature points we consider are peaks and peak-like protrusions of the terrain.

We make the following reasonable assumptions regarding the camera: 1) the height of the camera above the ground is known, and 2) the camera intrinsic parameters are known through a previous calibration procedure.

Let us make the following definitions. The *digital elevation map (DEM)* is an equidistant grid of elevation values. It contains the three dimensional coordinates of points on the ground.

The *actual skyline* is the sky-terrain boundary in an image. It is also the occluding contour of the earth's surface visible in the image. The *synthetic skyline* is the upper bound of the projection of all world points on the image plane.

A world point is called a *map feature point* if it is not occluded by neighboring world points when there is a small change in viewpoint. A *map peak* is a map feature point that is not occluded by neighboring world points even when there is a relatively large change in viewpoint. The projections of the map feature point and map peak on the image plane are called *image feature point* and *image peak* respectively.

The *position* of a camera is specified by giving its longitudinal, latitudinal, and elevation coordinates. We assume that the height of the camera above the ground is known so that the camera elevation coordinate is a function of its longitudinal and latitudinal coordinates. Camera *orientation* is completely described by giving its pan, tilt, and swing angles. The camera position and orientation together constitute the camera *pose*.

*presented during the 38th Research Meeting of the Pattern Sensing Group, Society of Instrument & Control Engineers, Tokyo, Japan, Jan. 31, 1997 (pp. 9 - 16)

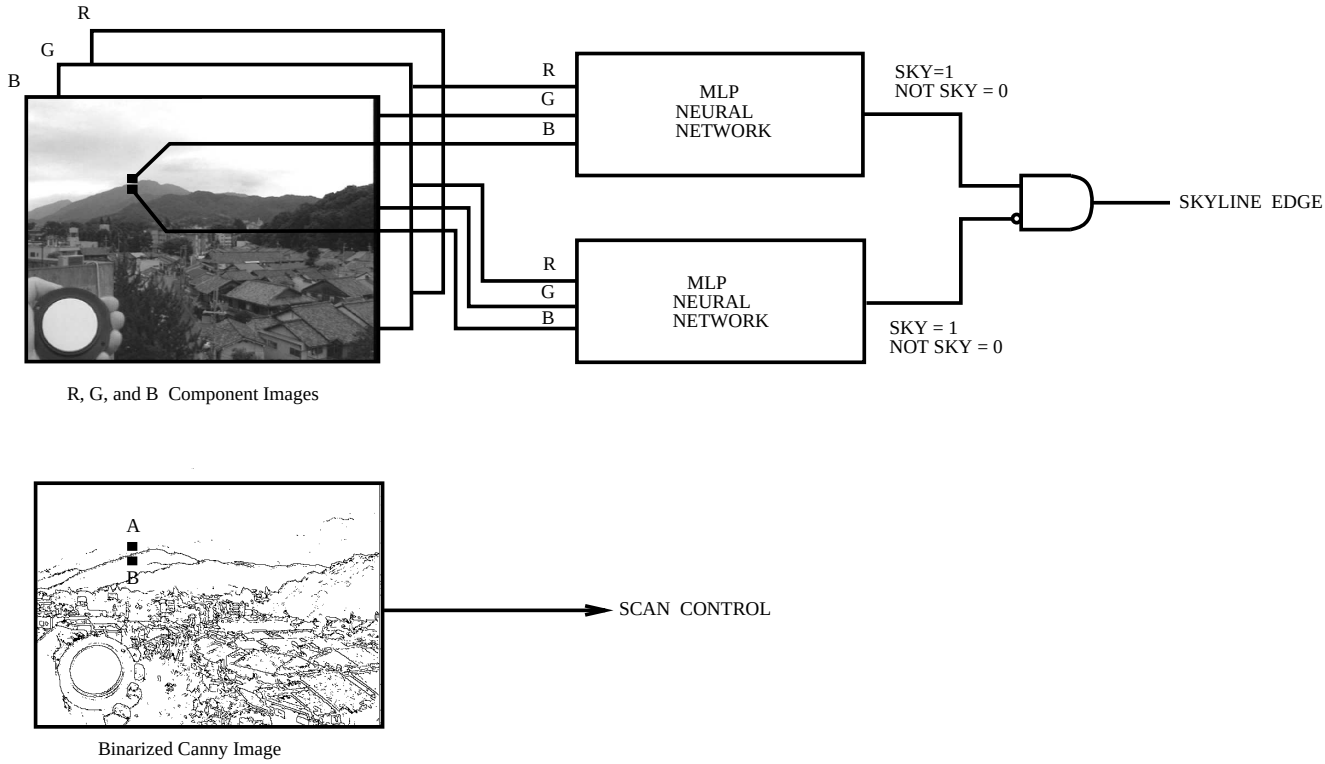

Figure 1: Neural Network-Based Skyline Extraction

2 Low-Level Processing

2.1 Skyline Extraction Using a Neural Network

Reliable extraction of the mountain skyline under the widely-varying lighting conditions of outdoor scenes is not a trivial task. Usual approaches to similar problems involve dynamic programming on the edges [6] or graph searching techniques such as A* [7]. Both DP and graph searching, however, require an evaluation function that measures the "skylineness" of an edge in order to obtain the maximand (in DP) or to guide the node expansion process (in graph searching). This evaluation function may be difficult to formulate.

We developed a neural network-based skyline extraction procedure that does not require an evaluation function. It relies instead on the color information in those pixels immediately above and below a skyline edge. We used a Multilayer Perceptron (MLP) to implement the neural network. Compared to both DP and graph searching, a neural network-based extraction module is easier to implement. The neural network must previously be trained to classify a pixel as either "sky" or "not sky".

The skyline is extracted in the following manner. First, the red component of the actual image is edge-detected using a Canny Filter. The red component was chosen because it provides the greatest contrast between a white sky and green mountains. The Canny Image is binarized using a low threshold value. The resulting Binarized Canny Image contains a lot of edges, only some of which constitute the mountain skyline (see Fig. 1). Skyline pixels are determined using the following procedure: Scan the Binarized Canny Image columnwise from top to bottom while looking for an edge pixel. Once an edge pixel is found, get the RGB values of a pixel just above it (call this pixel A) and those of another pixel (pixel B) just below it. Submit these RGB values to the neural network for classification. If A is classified as "sky" and B as "not sky" then label the edge pixel as a skyline pixel. Repeat until all the columns have been scanned.

The skyline is then obtained from the original Canny Image by contour following using the skyline pixels as a guide.

2.2 Image Feature Point Extraction

Accurate extraction of the image locations of image feature points is very important since these locations determine the pose. In this paper we only consider peaks and peak-like protrusions of terrain as feature points. In general, peaks do not have a strictly defined shape and the shape of a particular peak changes with viewpoint. We can, as a first approximation, model a peak as a Gaussian and use

the characteristics of the Gaussian curve in searching for peaks in the image. The idea of using the Gaussian as a model for terrain was used by Lavin [8] in scene analysis. We describe below a peak extraction procedure based on the search for Gaussian characteristics in the second derivative of the image skyline. We used a Savitzky-Golay Differentiating Filter [9] to obtain the skyline's second derivative.

The Gaussian curve can be described by the equation $g(x) = exp(-x^2/2\sigma^2)$ where σ is the value of x at which the function value falls to $e^{-1/2}$ of its maximum (see Fig. 2). The Gaussian has the following properties:

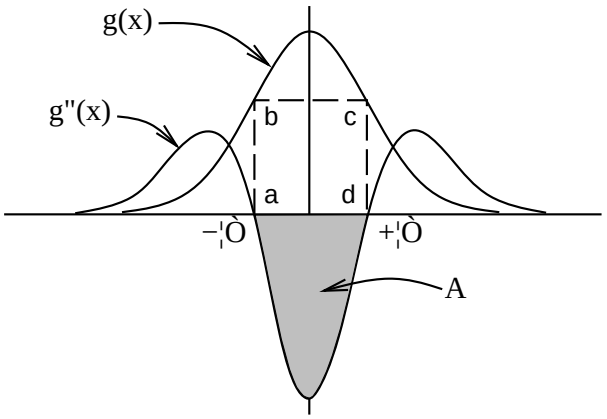

Figure 2: The Gaussian and Its Second Derivative

1. the second derivative of the Gaussian function has its zero crossings at $x = \pm\sigma$,
2. that the Gaussian function is maximum when its second derivative is minimum,
3. the area bounded by the x-axis, and the second derivative curve from $-\sigma$ to σ is proportional to the area *abcd* under the Gaussian function., i.e.

$$A = \int_{-\sigma}^{\sigma} -g''(x)dx = xg(x)/\sigma^2 \,|_{-\sigma}^{\sigma} \qquad (1)$$

The value A is a measure of how "large" the peak is. In the context of feature point extraction, prominent feature points will have large A's. We detect Gaussian-shaped mountain peaks in the image as follows: Scan the second derivative curve of the skyline for zero crossings. Find the intervals between zero crossings where the second derivative is negative. Compute the areas for each interval using eqn (1). Search the skyline for the highest points within the intervals (these are the candidate peaks). Rank the peak candidates according to their areas. Label the peaks whose areas are above a certain percentage (e.g. 30%) of the largest peak area as "peaks" and the remaining as "other feature points". Figure 3 shows the result of this procedure. Extracted peaks are ranked 1 to 7 in order of decreasing area.

This procedure performs relatively well even when compared to manual extraction. Occlusion, however, produces false peaks.

2.3 Map Feature Point Extraction

The earth's surface is a complex surface that is too complicated to be approximated by simple patches. It can, however, be decomposed into a union of fundamental surface types (peak, pit, ridge, valley, etc.) based on the mean and Gaussian curvature at each point. Since we will only consider peaks and peak-like protrusions as feature points, these points can be obtained directly from the DEM using a simpler procedure.

The elevation values in the DEM we used measure the *average* elevation of each 50m $\times$ 50m grid. The resolution of elevation values is 10 cm. There values were obtained from geodetic measurements so that they can be considered as noise-free. We can extract peaks and peak-like protrusions by searching for local maximas in the DEM. A peak can then be considered as the highest point in a small area where the area is the point's neighbors (e.g. 8-, 15-, 24-, etc. neighbors). We also need to specify a certain cutoff elevation value below which the search for peaks and peak-like protrusions is not performed. We store the locations of these points in a database.

The algorithm for generating a map feature point database is described below:

Scanning direction is from north to south, beginning at the northwestern-most point. When the southern-most point has been scanned, the next northern-most point is scanned next.

1. Label all points above a certain elevation with "candidate".
2. Check if there are still points labeled with "candidate". If there is none then go to 4. Scan and look for the points with the lowest value among those marked with "candidate". Call the first instance of the lowest value as D.
3. Examine the small area around D. Call all points in this small area as neighbors.

   3.1 If all the neighbors of D are labeled "not feature point" then label it "feature point", save its 3-D coordinates in the database and go to 2.

   3.2 If there is at least one neighbor that is higher than D, then label D with "not feature point" and go to 2.

   3.3 If there is at least one neighbor that has the same elevation as D then tag D with "feature point", save its coordinates into the database and mark all those neighbors with "not feature point".

4. Exit. The database has been made.

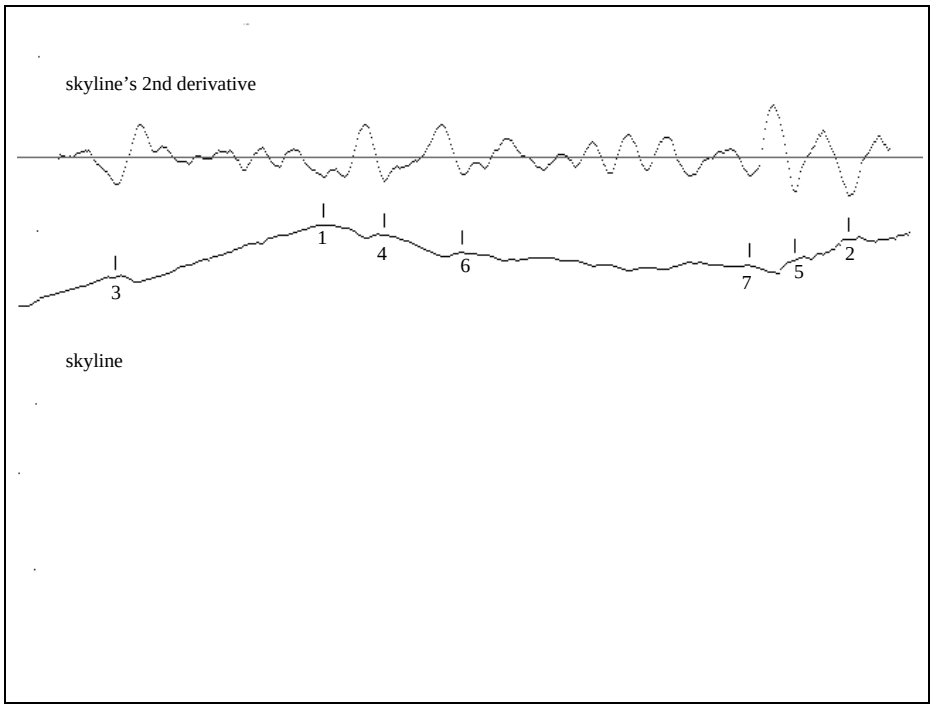

Figure 3: An Example of a Skyline and its Second Derivative

Our viewpoint estimation method requires two databases (the map peak database and the map feature point database). Ideally, the map peak database contains the prominent peaks and the map feature points database contains all the peaks (prominent or not) present in the DEM. These two databases are generated from the DEM using the algorithm described above but with different values of the small area parameter.

3 Computing Position and Orientation Using Nonlinear Least Squares

Solving for the camera pose from a given set of known image to model point correspondences can be formulated as a nonlinear least squares problem. We assume that the camera can be modeled using the simple pinhole camera model. If greater accuracy is desired, a more elaborate camera model can also be used. In this case, some minor modifications have to be made to the procedure we describe below.

The simplified pinhole camera model has one internal parameter (the focal length f), and six external or pose parameters, namely, the camera position variables $(x_{cam}, y_{cam}, z_{cam})$, and orientation parameters (pan angle θ, tilt angle ϕ, swing angle ψ). The position parameters specify the location of the focal point of the camera with respect to a world reference frame (Fig. 4).

A world point having coordinates $p = (x, y, z)$ is mapped into an image point $P = (u, v)$ according to the following perspective transformation equations:

$$(\hat{x}, \hat{y}, \hat{z}) = R(p - t) \tag{2}$$

$$P = (P_u, P_v) = (u, v) = (f\hat{x}/\hat{z}, f\hat{y}/\hat{z}) \tag{3}$$

where t is the translation vector describing the position of the focal point with respect to the world reference frame, $(\hat{x}, \hat{y}, \hat{z})$ are the camera-centered coordinates of the point, and R is the product of three rotation matrices, one for each axis.

Given n world point to image point correspondences, $p_i = (x, y, z) \leftrightarrow P_i = (u, v)$, $i = 1, 2, ...n$, we want to accurately compute for the six external parameters $\omega = (x_{cam}, y_{cam}, z_{cam}, \phi, \theta, \psi)$ that map p_i into P_i. Let $(P_u(p_i; \omega), (P_v(p_i; \omega))$ for $i = 1, 2, ..., n$, be the aligning transformation in Eqn. (3).

For an image taken by a real camera, noise and measurement errors, as well as deviations from the ideal pinhole camera model, make alignment impossible. Thus, we aim for minimizing the overall alignment error. Let us represent these errors by a residual vector

$$\begin{aligned} E(\omega) \quad = \quad & [P_u(p_1; \omega) - u_1, ..., P_u(p_n; \omega) - u_n, \\ & P_v(p_1; \omega) - v_1, ..., P_v(p_n; \omega) - v_n]^T \end{aligned}$$

To simplify notation, let $r_i(\omega)$ be the ith element of $E(\omega), i = 1, 2, ..., n$. The derivative of $E(\omega)$, called the Jacobian and denoted by $J(\omega)$, is an $m \times 2n$ matrix (m is the dimension of the vector ω) whose general element is $J(\omega)_{ij} = \partial r_i(\omega)/\partial \omega_j$. We compute

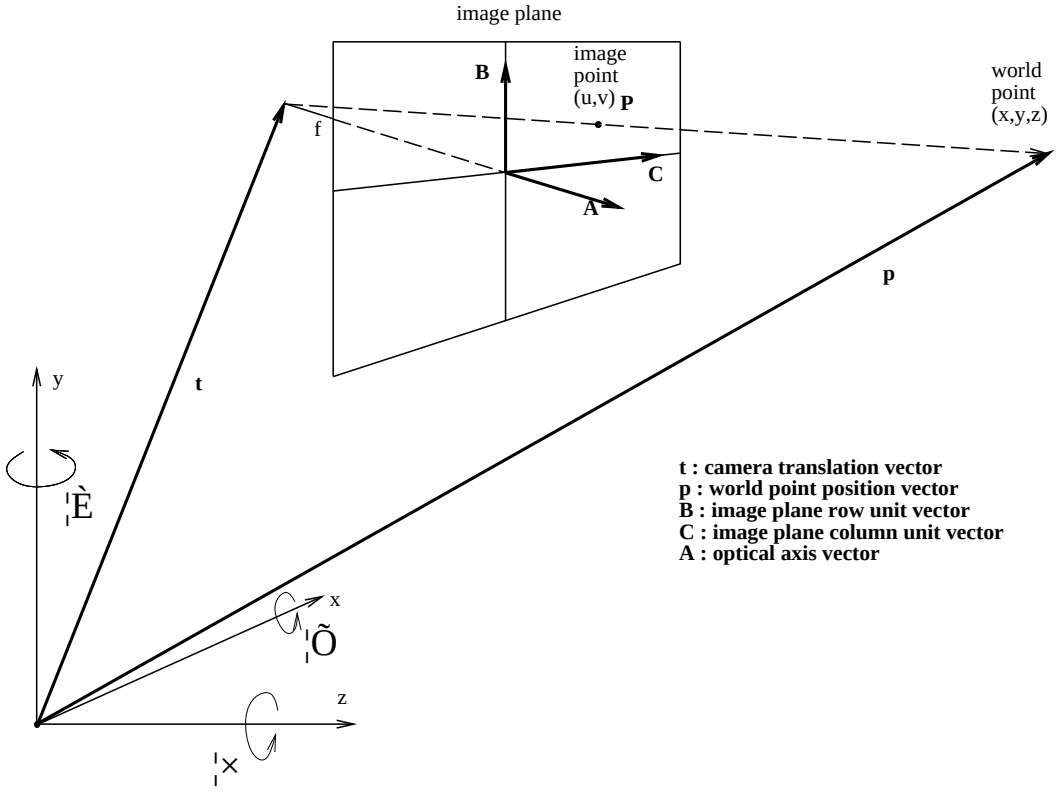

Figure 4: World Reference Frame

this Jacobian using its forward finite difference approximation since it is difficult to write an analytic expression.

Solving for the value of the camera parameters ω is the same as solving the nonlinear least squares problem

$$\underset{\omega}{minimize} \left\{ \| E(\omega) \| = \frac{1}{2} E(\omega)^T E(\omega) = \frac{1}{2} \sum_{i=1}^{2n} r_i(\omega)^2 \right\}$$

The nonlinear least squares problem above can be solved iteratively using the Levenberg-Marquardt Algorithm [10]:

$$\omega^{(k+1)} = \omega^{(k)} - (J(\omega^{(k)})^T J(\omega^{(k)}) + \mu^{(k)} I)^{-1} J(\omega^{(k)}) E(\omega^{(k)})$$

where $\mu^{(k)}$ is a nonnegative scalar the calculation of which is described in detail in [11]. Note that this procedure does not require an initial estimate of the parameter values.

For our problem, the value of $\| E(\omega) \|$ at convergence, called the residual, measures the degree of misalignment of the n points.

For world to image point alignment under perspective transformation, at least three point correspondences are required. However, the solution to the 3 world point to image point alignment problem is not unique. Fischler and Bolles [12] showed that there are at most four solutions. Wolfe et al [13] provided an in-depth analysis of this problem and explained why there are only two solutions in most cases. We circumvent this problem by putting a constraint on the camera elevation variable. Since the height of the camera above ground is known, the camera elevation is not an independent variable but is completely determined by its longitude and latitude. This constrains the solution to that consistent with the physical problem.

4 K-Nearest Feature Point Search

A minimum of three pairs of matching model and image points are needed to compute the pose using the nonlinear least squares optimization procedure described in the preceding section. The total number of hypotheses is $\mathcal{O}(n^3 m^3)$ for n map feature points and m image feature points. This value exceeds 10^{10} even for a small (30 km $\times$ 30 km) region.

An image typically captures one or more peaks together with several other feature points. Their corresponding map peaks and map feature points are close to one another in three dimensions. We make use of this observation to formulate our K-Nearest Feature Point Search Strategy.

Instead of hypothesizing the alignment of any three image feature points with any three map fea-

ture points, we impose the following unary constraints on the elements of the map feature point triple:

1. let the first map feature point in the triple be from the map peak database (let us call this the pivot peak).

2. let the next two feature points in the triple come from the set of k-nearest feature point neighbors of the pivot peak. This set may also include map peaks and is stored in the feature point database.

This strategy is very effective since the search for the correct pose is performed only on the search subspace where the pose is to be found.

A priori information can further reduce the size of the search subspace to be searched. If it is known beforehand that a particular mountain is visible in the image, then we can fix the assignments of the first point in the map feature point triple to the map peaks of that mountain. In this case, reduction in complexity by six orders of magnitude or more is possible.

5 The Inter-Feature Point Visibility Constraint

Further speed-up can still be achieved by eliminating invalid hypotheses using a constraint we call Inter-Feature Point Visibility Constraint. This binary constraint is based on the following observation. If two image feature points are "visible to each other", then their corresponding map points must also be "visible to each other". If this constraint holds for a pair of image feature points and their corresponding model points are blocked by intervening matter, then the hypothesis is invalid and can be rejected without further computation. Whenever valid, this binary constraint is applied on the three possible pairings of elements of the image feature point triple.

6 Verification of Hypotheses

Hypotheses that survive the constraints above (consistent hypotheses) are submitted to the Levenberg-Marquardt Nonlinear Least Squares Optimization module so that their corresponding poses can be computed. The nonlinear least squares optimization procedure filters out hypotheses that cannot possibly cause image to model feature point alignment. The amount of misalignment for those that were aligned is given by the residual. Poses with residuals larger than a specified small value are rejected.

Ideally, for a correct pose, each model point must project either on the image skyline itself or below it but never above it. We make use of this fact to eliminate other impossible poses. Thus we have the following constraint: Project model points onto the image plane while checking whether projections are on or below the skyline. If one model point projects above the skyline (plus a small margin to take into account image noise and other errors), eliminate the pose. We call this the "Skyline is the Limit" Constraint.

The remaining poses constitute the candidate poses. Synthetic skylines for each pose are generated. These synthetic skylines are then ranked according to their degree of match with the actual skyline. The pose whose skyline closely matches the actual skyline is then selected as the best pose.

This viewpoint estimation by alignment method can also tolerate partial occlusion. Occlusion has the effect of increasing the number of hypotheses (which will eventually be rejected) since the low-level processing stage extracts "false feature points" from the occluding matter. This can be lessened by putting some form of mountain/occluding matter recognition capability in the low-level processing module.

7 Experiments

We tested our method on 32 images of a mountain (Hieizan near Kyoto City) taken from 11 different locations using a portable video camera. A tripod was not used. Simulations using synthetic images showed that positional accuracy is not significantly affected by small swing angles ($\psi < 10°$).

The map peak database and map feature point database were generated using 224- and 48- neighbors respectively, and a cutoff elevation of 300 meters. We used a 15-nearest feature point search.

The total number of image feature points extracted varied from 3 to 15 but only the top 6 were considered to reduce the number of hypotheses. Position errors and processing times on a 170 MHz Sun Ultra Workstation are given in the tables below. Processing time does not include low-level processing. For a priori information, we used the fact that Hieizan is present in the image so that only the two most prominent peaks of the mountain were considered as pivot peaks.

Position Errors (meters)		
Min	Mean	Max
90	373	980

Processing Time (min:sec)			
A Priori Info	Min	Mean	Max
none	14:30	1 hr 36 min	4 hr 55 min
present	0:35	4:28	13:44

The percentage of hypotheses rejected by the Inter-Feature Point Visibility Constraint is highly variable (7% to 42%) depending mainly on viewpoint and presence of occlusion.

Camera orientation results are difficult to quantify but they can be evaluated qualitatively by visual inspection. Figure 5 shows one result for an image that has partial occlusion. The black curve is the model skyline projected on the original mountain image using the position and orientation values obtained by the method.

We expect an improvement in positional accuracy if a camera model that takes into account distortion and other errors is used. The position estimate can still be refined by using more than three image point to model correspondences in the nonlinear least squares optimization process. Future work will address these issues.

8 Conclusion

We presented a method for estimating the camera position and orientation parameters from a single mountain image given a digital elevation map. It follows the object recognition alignment scheme of model-based vision. We also discussed a technique for avoiding combinatorial explosion. Experimental results show that the method is feasible.

References

[1] W.B. Thompson, H.L.Pick Jr, B.H. Bennett, M.R. Heinrichs, S.L. Savitt, and K. Smith, "Map-based localization: The 'drop-off' problem", *Proc. DARPA Image Understanding Workshop*, 1990.

[2] M.D. Ernst and B.E. Flinchbaugh. "Image/map correspondence using curve matching", *AAAI Symposium on Robot Navigation*, pp. 15-18, March 1989.

[3] A.C. Kak, K.M. Andress, C. Lopez-Abadia, and M.S. Caroll, "Hierarchical evidence accumulation in the PSEIKI system and experiments in model-driven model robot navigation," in *Uncertainty in Artificial Intelligence (Vol. 5).* Amsterdam: Elsevier, 1990, pp. 353-369.

[4] R.. Talluri and J.K. Aggarwal, "Image/map correspondence for mobile robot self-location using computer graphics", *IEEE Trans. Patt. Anal. Mach. Intel.*, vol. 15, no. 6, pp. 597-601, 1993.

[5] R. Nevatia, K. Price, and G. Medioni, "USC Image understanding research: 1990 - 1991", *Proc. DARPA Image Understanding Workshop*, 1992.

[6] D.H. Ballard and C. M. Brown, *Computer Vision.* Englewood Cliffs, NJ:Prentice-Hall, 1982, pp. 131-136.

[7] Martelli, A., "An application of heuristic search methods to edge and contour detection" *Commun. ACM 19,2,* Feb. 1976, 73-83.

[8] M.A. Lavin, "Analysis of scenes from a moving viewpoint", In *Artificial Intelligence: An MIT Perspective, Vol.2* P.H. Winston and R. H. Brown (editors), pp. 185-207, MIT Press, 1979.

[9] A. Savitzky and M.J.E. Golay, *Anal. Chem.* vol. 36, pp. 1627-1639.

[10] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Englewood Cliffs, NJ:Prentice-Hall, 1983.

[11] J. J. More, "The levenberg-marquardt algorithm: Implementation and theory", in Numerical Analysis, G. A. Watson (ed) *Lecture Notes in Mathematics 630*, Springer-Verlag, 1977.

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Commun. ACM*, vol.24, no. 6, June 1981.

[13] W. J. Wolfe, D. Mathis, C. W. Sklair, and M. Magee, "The perspective view of three points", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-13, no. 1, Jan. 1991.

Figure 5: Result for an Image with Partial Occlusion