Stealth Navigation: Planning and Behaviors*

Srinivas Ravela, Richard Weiss, Bruce Draper, Brian Pinette
Allen Hanson and Edward Riseman
Computer Vision Research Laboratory
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

Abstract

Stealth navigation is a fundamental capability for a scout vehicle to effectively perform reconnaissance missions. One component of stealth is the ability to plan paths that minimize visibility of the vehicle from target locations. This paper demonstrates a stealth planning system using visibility analysis on digital terrain elevation maps for path planning. Goal locations can be user specified or automatically constructed at the visibility boundary for RSTA operations. The second component of our stealth navigation system is the ability to control a vehicle to minimize exposure during RSTA operations in the presence of sensor measurement error, inaccurate terrain maps or a dynamic environment. The notion of visual stealth behavior, i.e. visual servoing to acquire and maintain a stealth position or trajectory is introduced as a mechanism for robust stealth navigation. Matching and tracking terrain features in the image space are identified as important pre-requisites for visual stealth behaviors. A method is described to track an important terrain feature in image-space namely, horizon curves.

1 Introduction

An important capability for a scout vehicle is *stealth navigation* or the ability to navigate with a minimum exposure from a target. This is important for establishing reconnaissance, surveillance and target acquisition(RSTA) positions during military operations. Where such stealth is impossible and the vehicle must be exposed to accomplish its mission, partial cover is desirable.

More formally, let us consider a *stealth condition* to exist if the target is visible by the sensor on the vehicle while the vehicle itself remains hidden. One requirement for stealth navigation is the ability to plan and execute stealth paths derived from terrain maps. These paths may be planned to pre-specified locations or to locations where the stealth condition becomes true. A second important requirement for stealth navigation is to generate modes of vehicle behavior from terrain cues that maintain or make the stealth condition true.

The motivation for using terrain features lies in the need for robust stealth navigation, i.e. to be able to react to uncertainty in the environment arising from inaccurate terrain maps, inaccurate sensory modalities such as odometry or to react to a dynamic environment where the target is moving.

In this paper we present a method for planning paths that are not visible from the target using an accurate reconstructed terrain map. In addition, we develop some of the components necessary for *visual stealth behavior* i.e. the use of visible terrain cues to generate vehicle behavior, with the goal of maintaining or making the stealth condition true.

Digital terrain elevation maps [Schultz94] are used to plan a stealth path given a starting point, target locations and a destination. Path planning is accomplished using harmonic potential functions [Connolly90] over a discretized local region to avoid locations that are a) visible from the target (visible-to-target) or b) untraversable according to the terrain elevation map (terrain obstacles). Path planning is traditionally posed in terms of obstacle avoidance, but here both visible-to-target and terrain obstacle locations are treated as obstacles by the planner (planner obstacles). The planner produces smooth obstacle-free paths to the goal if such solutions are possible. The planning algorithm is robust in the presence of environment model errors (such as un-modelled obstacles) and provides a framework for incrementally acquiring environment models (see section 3). VLSI prototypes of the planner have been designed [Stan94] for generating path plans in real-time. Thus, paths can be planned in real-time, within safe regions and at both the mission and execution levels. In addition alternate

*This work was funded in part by TACOM DAAE07-91-C-RO35, NSF IRI-9208920, and NSF IRI-9116297.

paths around un-modelled obstacles can be dynamically computed whenever visual sensors detect such obstacles.

Visual stealth behaviors operate by transforming visual observations of terrain features into motor commands to the vehicle. For example, a fundamental visual stealth behavior is maintaining a stealth condition relative to the target, by servoing the vehicle to interpose an occluding contour of a hill, tree or building. Another type of visual stealth behavior uses horizon curves, an occluding contour, and knowledge of the approximate location of the target to servo the vehicle just to the point of visibility and to maintain the stealth condition from then on (see Figure 3).

Some of the best terrain features for visual stealth behavior are distinctive horizon lines and occluding contours of hills. These features can be used in several different stealth behaviors. Horizon curves can be used for visual servoing to control orientation. Horizon curves and occluding contours can be used together to predict the visibility of the target and scout vehicle, and as part of visual servoing to keep the vehicle on the boundary of visibility. A prerequisite for this type of visual servoing is the ability to match and track curves over a sequence of images as the vehicle is moving.

The basic components of tracking and visual servoing are not new. In order to show that this approach is feasible, we employ synthetic data to show that matching and tracking can be done on horizon curves. Tracking can be incorporated into real-time visual servoing and while we have not demonstrated this part in the context of outdoor navigation, there is a working system which uses these techniques in the domain of automated assembly [Ravela94].

2 Related Work

A complete review of literature covering navigation, planning, feature extraction, matching and tracking is beyond the scope of this paper. Instead, we provide pointers to more detailed information. Khatib promoted potential fields for path planning [Khatib85]. For a detailed survey of harmonic functions see [Connolly93] and for a thorough introduction to path planning see [Latombe91]. For a survey of architectures for mobile robot navigation, including planning and reactivity see [Ravela92]. The kernel of the tracking system described in this paper is based on normalized correlation [Fennema91] in polar space. For studies of correlation and SSD see [Wood83]and [Anandan89] respectively.

Tracking has been approached from a token-based view point and [Deriche91, Koller93] where the tokens are line segments or groups of segments. Direct methods for contour tracking were employed in [Kass88]; for a comprehensive study see [Blake92]. These track contours by associating a deformation model with a contour network. Model based approaches have also been studied [Gennery82, Verghese90]. Correlation based approaches are seen in [Hager94] and for a good study on feature based method see [Shi94]. Our tracking technique is close to [Hager94] and is hybrid in that both image based and edge information are employed. Further, instantiation of the tracker is model-based and tokens can be tracked by grouping these features appropriately. Unlike the approach presented by Hager, we employ polar space correlation to compensate for large instantaneous rotations.

An excellent study of visual servoing can be found in [Hashimoto94], which primarily covers a manipulator, hand-eye environment. Visual servoing has been used on mobile robots in structured environments [Zhang93] and in outdoor environments [Fennema91, Dickmanns88]

3 Stealth Planning

Stealth paths which avoid terrain obstacles and visible-to-target locations are planned over a region Ω of the terrain map using harmonic functions. Terrain obstacles are identified as areas in Ω which are not traversable by the vehicle and can include steep slopes, rivers, lakes, heavily forested areas etc. A point in the discretized region Ω is considered to be visible-to-target if the line joining the point and the target location (after suitably accounting for the height of the vehicle) does not intersect the elevation surface over Ω. As noted earlier, the set of planner obstacles is a union of the set of target visible and terrain obstacle locations.

A function in the domain $\Omega \subset R^n$ ($n = 2$ in our case) is harmonic if it satisfies Laplace's equation:

$$\nabla^2 \phi = \sum_{i=1}^{n} \frac{\delta^2 \phi}{\delta x_i^2} = 0$$

In the context of this paper the 2D cartesian region Ω is also the configuration space of a holonomic robot[1]and its boundary $\delta\Omega$ contains the boundaries of all obstacles and goals. Assuming Ω is represented by a regular grid, the goal is held at potential $\phi = 0$ and the cells in the grid corresponding to $\delta\Omega$ are held at obstacle potential $\phi = 1$. A harmonic function over Ω is computed as a discrete relaxation over the free cells of the grid, where at each iteration a cell is replaced by the average of its city block neighbors. The relaxation process terminates when

[1] the planner works in the configuration space of this holonomic robot

there is no change in any cell from one iteration to the next. Once the relaxation terminates, paths (or streamlines) may be generated by following gradients from the start point. Multilinear interpolation may be used to interpolate between grid nodes, since functions generated thereby are harmonic. See [Connolly93] for an analysis of numerical methods to compute the function ϕ.

Harmonic functions have a number of important properties:

⋆ Every Harmonic function is analytic and lacks spurious local minima [Connolly90].

⋆ Completeness up to discretization error: In addition to the aforementioned assumptions for computing ϕ, if it is assumed that Ω is a compact subset of R^n, then the path planning scheme is complete up to the approximation of the environment [Connolly93].

⋆ Provides a framework for extension and refinement of environment models: Upon sensing a new obstacle, the configuration space map can be updated incrementally when Jacobi iterations are used for relaxation. See [Connolly93] for a discussion.

⋆ Robustness in the presence of unanticipated obstacles and errors: Every streamline is obstacle free and must lead to a goal, if the environment has been modelled accurately. If the current streamline is blocked due to an un-modelled obstacle, then alternate streamlines may be picked. Connolly and Grupen [Connolly92, Connolly93] describe a method to generate robust trajectories under model errors.

⋆ Behavioral variability through superposition: The algorithm for computing ϕ assumes Dirichlet-style boundary conditions. Under these conditions streamlines tend to *repel* away from the obstacles. Tarrasenko and Blake [Tarassenko91] use Neumann-style boundary conditions resulting in streamlines that tend to *graze* obstacles. Since the superposition of harmonic functions is harmonic, the functions obtained from these two boundary conditions can be mixed to generate different modes of behaviors, ranging from paths that graze obstacles to paths that are repelled away from them.

⋆ Rapid Computation: Harmonic function computations may be computed using resistive networks. A VLSI implementation of this scheme is in the prototype stage and analog implementations are discussed in [Stan94]. It is anticipated that grids of modest size (100 by 100) will be able to compute harmonic functions within 100 microseconds [Connolly93].

Figure 1: Planning a stealth path from start (S) to a goal (G) on a terrain reconstructed from the DemoB site.

3.1 Experiments

The terrain over which both these experiments were conducted are the Martin-Marietta Demo B site. Please note that the standard $30m$ DTED could be used, but we have used the high resolution elevation maps from the stereo reconstruction algorithms described in [Schultz94].

Figure 1 shows a planned stealth path from start (S) to a specified goal location(G). The target(T) is located at the upper left corner on the building at the bottom of the figure. The discretized region Ω over which the path was planned is shown as a regular grid of vertical black dashes. The width of the grid Ω is two thirds of the distance between the start and the goal plus a clearance of 30 pixels on either side. Planner obstacles are marked in grey while the path from start to the goal is marked in white and the image in the background is the terrain image. In this case, both the Neumann and Dirichlet boundary conditions were used to compute harmonic functions (ϕ_n and ϕ_d respectively) and the graphed path is a result of steepest descent over an equal mix of ϕ_n and ϕ_d. Note that there may be no path within the local region encompassing the start and goal locations. From the completeness property this condition can be detected. Therefore, grid sizes can be incrementally increased via an interactive planner till a path is found.

Figure 2 shows an example of a class of stealth plans, where a path is planned to a visibility boundary for RSTA, given a start and target location. The goal in this case is any location that is not a terrain obstacle, lies at a boundary of a visible region, and is not visible. The dis-

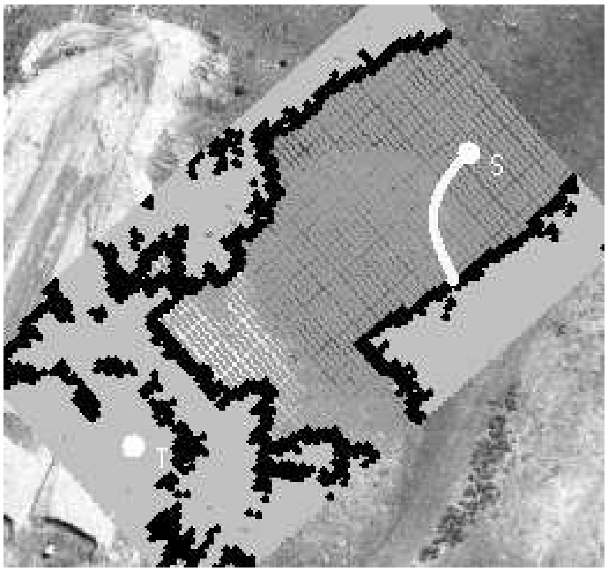


Figure 2: Planning a path for RSTA from (S) to a location on the visibility boundary(shown in black) of a target (T).

cretized region Ω is marked as a regular grid of white dots. The grid parameters are the same as above, except that the grid length is constructed from the distance between the target(T) and the start(S). The visible region is painted in grey and the candidate goal locations on the visibility boundary are the set of black lines. Again a weight of 0.5 was used on the functions computed from the two boundary conditions, and the steepest descent path is marked in white. Note that while there is a shorter path to a goal location on the visibility boundary bearing immediately left of the start location, the actual path taken is the one of steepest descent in the direction shown by the thick white line in Figure 2. This is due to the fact that boundary locations are held at the obstacle potential and hence the path first starts straight out and then turns smoothly. The relaxation process results in a much steeper gradient in that direction than the geometrically shortest path.

4 Stealth Behaviors

In section 1, the notion of visual stealth behaviors was introduced. Our approach is to translate this into a general visual servoing paradigm. There is a strong motivation for the use of visual servoing in stealth, especially when other sensory modalities such as odometry are inaccurate, or when the the environment model becomes inaccurate. As position errors due to odometry inaccuracies accumulate, visual servoing from distinctive landmarks can be used to reduce these errors to manageable levels [Fennema91, Zhang93].

In a stealth scenario, for example, when the ve-

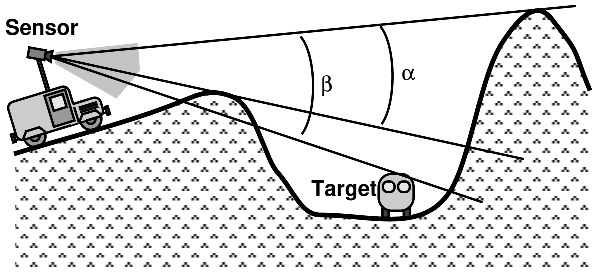


Figure 3: Establishing a stealth condition when approaching a target, by tracking terrain features

hicle is approaching a boundary of visibility, inaccurate odometry or terrain maps might cause it to become prematurely visible to the target, violating the stealth condition. However, if the approximate distance between the target and a distinctive horizon line is known *a priori* and if the horizon line and the occluding contour can be identified, then servoing information is available. As illustrated in 3 the target would become visible when the angle α subtended on the principal point of the camera by the chord joining the horizon and occluding contour becomes greater than or equal to the angle β subtended by the chord joining the horizon line and the target. By tracking the occluding and horizon contours the error between α and β can be computed continuously. This error can be converted to a motor command that servos the vehicle to the appropriate position (*approach* behavior). Once the target has been identified, it may be desirable to maintain the vehicle in a stealth condition relative to the target. This may be achieved by servoing the vehicle so $\alpha - \beta$ stays below a limit determined by the height of the sensor platform from the top of the vehicle(*stance* behavior).

Approach and *Stance* are two simple behaviors that rely on visual servoing from image-space feature tracking to acquire and maintain stealth. We believe them to be elements of a broader repertoire of possible stealth behaviors. In particular, the reconstructed terrain will provide other nearby terrain features, whose image space appearances can be used as a basis for servoing.

Next we demonstrate the initial effort in the implementation of such simple behaviors by showing how matched curves can be tracked. Then we describe experiments with matching horizon curves using footprints [Kalvin86], and tracking them over a motion sequence.

4.1 Tracking

In order to track object features in the image, first a correspondence between the object features and their image space appearances must be obtained. This can be used to construct or create templates that capture the feature. Then these templates must be registered over a temporal sequence of images. First a technique for template registration using a combination of normalized correlation [Fennema91] and steerable filters [Freeman91] is developed. Then experiments with horizon curve matching and tracking are presented.

Normalized correlation of a template $\tau(x,y)$ over a set

$$T = \{(x,y) \,|\, -t_x \leq x \leq t_x, -t_y \leq y \leq t_y\}$$

with an image patch $\gamma(x,y)$ over the set

$$I = \{(x,y) \,|\, -i_x \leq x \leq i_x, -i_y \leq y \leq i_y\}$$

at a location $-i_x + t_x \leq i \leq i_x - t_x, -i_y + t_y \leq j \leq i_y - t_y$ is given in a computationally efficient form by

$$\frac{2 * \sum\limits_{m,n} \tau(m-i, n-j) * \gamma(m,n)}{R1 * \sum\limits_{m,n} \tau(m-i,n-j)^2 + R2 * \sum\limits_{m,n} \gamma(m,n)^2} \tag{1}$$

$$R1 = \frac{\sum\limits_{m,n} \tau(m-i, n-j)}{\sum\limits_{m,n} \gamma(m,n)}$$

$R2 = \frac{1}{R1}$ and $i - t_x \leq m \leq i + t_x$, $j - t_y \leq n \leq j + t_y$.

Theoretically, this measure assumes that the surfaces in the environment are Lambertian, that they can be locally approximated by a plane, and that the illumination incident on the surfaces can be locally approximated by a constant. Under these assumptions the correlation measure is normalized in that it is independent of the illumination incident on the surface. However, good experimental results have been obtained with this measure on surfaces that are only weakly Lambertian (see [Fennema91] for a derivation and [Fennema93] for experiments with this measure). While this method is useful in tracking features captured as templates over a temporal sequence of images, its performance degrades when the features undergo a rotation. In order to compensate for 2D rigid rotation it is sufficient to observe that equation 1 is linear shift invariant but is formulated in cartesian space. Since linear shift invariance in polar space is equivalent to rotational invariance in cartesian space, we formulate an equivalent correlation expression in polar space and the following sequence of steps are followed:

The templates used here are pairs $\langle \tau, \theta_t \rangle$ where τ is an image patch centered over a dominant image edge and $\theta_t \epsilon [0,\pi)$ is the orientation response of the steerable filter with the edge at the patch center.

To localize a template within a search window γ in a new image the following steps are followed:

1. Spatial gradients and their orientations are computed by filtering γ with steerable Gaussian derivative filters and suppressing non-maximal edges within the search window.

2. Each local maximal edge location (i,j) in γ is a potential candidate for the new location of the template, and normalized correlation in polar space is used to identify the best match.

Normalized correlation (described in equation 1) is performed over a cartesian parameterization of the sampling sequence. In polar space this is transformed into the radius(ρ) and angle(θ) parameters. The expression in polar space equivalent to the cross term in equation 1 is given by:

$$2 * \sum_{\rho=0,\theta=0}^{\frac{x^2}{2},2\pi} \tau(m_0, n_0) * \gamma(m_1 + i, n_1 + j)$$

where $m_0 = \rho * cos(\theta + \theta_t)$, $n_0 = \rho * sin(\theta + \theta_t)$,$m_1 = \rho * cos(\theta + \theta_i)$, $n_1 = \rho * sin(\theta + \theta_i)$ and $\theta_i \epsilon [0, \pi)$ is the orientation response at (i,j). The equivalent term for the denominator in equation 1 can be determined similarly.

Often for a motion sequence the motion is approximately known and new template locations can be hypothesized and localized using polar correlation and steerable filters as described above in appropriate search windows around the hypothesized locations. In case there are no motion estimates (but the motion is constrained under certain dynamics), localization can be performed in appropriate search windows around the current template location. See [Ravela94] for a discussion of both cases.

The advantage of using steerable filters is that they can be represented as a set of basis filters from which the orientation that maximizes the contrast of an edge can be analytically determined. Its performance is better than box filters, for example, when there are non-step edges. While polar correlation compensates for any changes in orientation of the feature there is, however, an issue of sampling and interpolation accuracy when going from cartesian coordinates of the image to the polar coordinates under which normalized correlation is performed. Accuracy is traded for speed to a certain degree in the real-time applications we have investigated, and sampling is performed only up to a pixel accuracy.

4.2 Experiments

The tracking method described here has been used for visual servoing of modelled objects manipulated by a six degree of freedom robotic manipulator with inaccurate kinematics. The tracking system ran at speeds up to 8Hz(See [Ravela94]). Here we describe tracking in the context of horizon curves over 145 rendered images of a $56m$ motion sequence on a terrain reconstructed using algorithms presented in [Schultz94]. The steps involved are presented below:

1. Model Curve Extraction: Given a continuous elevation surface, an occluding contour with respect to a camera may be defined as locations on the surface where the rays from the camera are tangential. A horizon curve is an occluding contour with the additional constraint that the tangent rays do not cut the elevation surface anywhere else. In computing the model curve we used the following approximation of this definition: curves, represented as point lists are extracted out of an image by first computing the edge locations using an edge operator such as steerable Gaussian derivative filters and then linking them based on proximity and continuity of orientation. With some user input, appropriate curves are picked as horizon curves. Figure 4(i) shows the extraction of two horizon curves from the first rendered image out of 145.

2. Data Curve Extraction: In this experiment data curves were extracted from a rendered image by computing the edge locations and then linking them using the aforementioned technique. Note that there is no user input in this case. Figure 4(ii) shows the raw data curves extracted from the 46th image in the sequence. Between the first and forty sixth images the camera has undergone a translation primarily into the scene by a distance of approximately $16m$.

3. Matching: As a pre-processing step data curves that are outside a nineteen pixel neighborhood of the model curve are pruned. This value is obtained by assuming a 2^o error in the model curve projection. Then, extremely small curves (less than five pixel wide) are pruned and the remaining curves are candidates for matching. Matching is accomplished using footprints as described in [Kalvin86] and curves that obtained a raw match score beyond a threshold (indicating at least a 50% match) are depicted in Figure 4(iii). Model curves are marked in black and the four data curves in white (the hill on the left in Figure 4(iii) has two data curves). Note that it is quite possible in general that "false" matches can occur i.e. data segments similar to the model in curvature may inhabit different parts of the image especially for low curvature model segments. In order to reduce the number of mismatches, only data curves that attain a high match score and are mutually consistent in rotation and translation (within a certain tolerance) are picked as the final matched data and tracking is performed on these segments.

4. Template Initialization: Figure 4(iv) illustrates template initialization indicated at the locations of the square white boxes. As noted earlier, the leftmost data curve was automatically pruned out since the rotation and translation parameters computed with the left model curve in Figure 4(iii) was inconsistent with the others, on which templates are initialized. Templates are initialized as follows; High curvature points that are represented as long lines in the 45^o or -45^o direction of the turning angle graph [Kalvin86] of the data segments are used as initialization points. In case there are no turns in the turning angle graph then the given segment has very low curvature and can be approximated as a straight line and two templates suffice. Otherwise the groups of templates represent vertices of a piecewise linear approximation of the curve. Once the templates are initialized, they are tracked using the tracking technique described in section 4.1.

5. Tracking: Figure 4(v) is approximately 40 frames after the template initialization frame in figure 4(iii)(representing $16m$ camera motion approx.). The camera motion during this is a curve to the left and consequently the templates appear shifted right in Figure 4(v). Figure 4(vi) is a continuation of the tracking along the same arc, spaced approximately 40 frames (16 m) from the the previous snapshot. The last two feature templates are lost because the features they were tracking have disappeared.

5 Conclusions

It is important that a scout vehicle be able to navigate with minimum visibility from a target. One component for achieving stealth is by planning stealth paths over *a priori* information obtained from the terrain map. The planner described in this paper produces smooth paths that are obstacle free and not visible to the target. Further, the planner is robust under environmental uncertainty, operates in real-time and provides a framework for incremental model acquisition.

A second component is the use of visual behaviors to control the vehicle for stealth during RSTA operations. One of the most fundamental capabilities for visual stealth behavior is the ability to match and track terrain features. We have demonstrated the use of a tracker that is based on rotation compensated normalized correlation and is amenable to real-time operation. We have shown an example where we have applied an algorithm [Kalvin86] to match and track horizon curves.

6 Future Work

The planning algorithm provides a framework for robust trajectory generation in the presence of errors such as un-modelled obstacles. We have developed a real time stereo-based obstacle detection system that registers obstacles on a local map[Badal94]. This can be used to trigger the planner to look for alternate streamlines when current ones are blocked, similar to the "sidetracking" mechanism described in [Connolly92]. Second, the paths constructed here are for a point, unit mass holonomic vehicle. Extensions to the planner have been made to account for the mass and non-holonomic nature of the HMMWV for example and field tests have to be conducted with these modifications. In addition a VLSI chip is being developed to allow path planning to take place in less than 100 microseconds. We are conducting indoor experiments with model registration under unknown camera motion using the tracker described here. Experiments with a real vehicle and imagery are to be scheduled to test a servoed system in an outdoor scenario. Most importantly it is of interest to design and integrate a broad repertoire of stealth behaviors using visual cues such as horizon lines, occluding contours of hills, trees, buildings and other types of cover.

Acknowledgements

The authors wish to thank Chris Connolly for providing the harmonic function path planner and Howard Schultz for providing elevation data, terrain pictures and the rendering software.

References

[Anandan89] Anandan, P., "A Computational Framework and an Algorithm for the Measurement of Visual Motion", *Int. J. Comput. Vision*, 2:283-310, 1989.

[Badal94] Badal, S., Ravela, S., Draper, B., Hanson, A., "A Practical Obstacle Detection and Avoidance System", *to appear, Proc. IEEE workshop App. of Comput. Vision*, 1994.

[Blake92] Blake A., and Yuille, A., (ed.), "Active Vision", *MIT Press*, 1992

[Connolly90] Connolly, C. I., Burns, J. B. and Weiss R., "Path Planning Using Laplace's Equation", *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2102-2106, May 1990.

[Connolly92] Connolly, C. I. and Grupen, R. A., "Harmonic Control", *Proc. Int. Syp. Intell. Control*, Glasgow, Scotland, UK, Aug 11-13, pp. 503-506, 1992.

[Connolly93] Connolly, C. I. and Grupen, R. A., "The Applications of Harmonic Functions to Robotics", *J. Robotic Systems*, 10(7):931-946, Oct. 1993.

[Deriche91] Deriche, R., and Faugeras, O., "Tracking line segments", *Image and Vision Computing*, pp 261-270, 1991.

[Dickmanns88] Dickmanns, E. D., and Graefe, V., "Dynamic Monocular Machine Vision", *Machine Vision and Applications*, 1:223-240, 1988.

[Fennema91] Fennema, C. L., "Interweaving Reason, Action and Perception", *COINS TR91-56*, Dept. of Computer Science, Univ. of Massachusetts, Amherst, 1991.

[Fennema93] Fennema, C. L., "Finding Landmark Features Under a Broad Range of Lighting Conditions", *Proc. SPIE Intelligent Robots and Computer Vision X11: Algorithms and Techniques*, 2055:181-191, Boston, MA, Sept. 1993.

[Freeman91] Freeman, W. T. and Adelson, E. H., "The Design and Use of Steerable Filters", *IEEE Trans. Patt. Anal. Machine Intell.*, 13(9):891-906, Sept., 1991.

[Gennery82] Gennery, D., "Tracking known 3D objects", *Proc. conf. AAAI*, pp13-17, 1982.

[Hager94] Hager, G. D., "Real-Time feature tracking and projective invariance as a basis for hand-eye coordination.", *Proc. Comput. Vision Patt. Recognition*, pp. 533-539, 1994.

[Hashimoto94] Hashimoto, K., (ed.), "Visual Servoing", *World Scientific*, 1994.

[Kalvin86] Kalvin, A., Schonberg, E., Schwartz, J. T. and Sharir, M., "Two-Dimensional, Model-Based, Boundary Matching Using Footprints", *Int. J. Robotics Research*, 5(4):38-55, 1986.

[Kass88] Kass., M., Witkin, A. and Terzopolous, D, "Snakes: Active contour models", *Int. J. Comput. Vision*, 1(4):321-331, 1988.

[Khatib85] Khatib, O., "Real-time obstacle avoidance for manipulators and mobile robots", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 500-505, 1985.

[Latombe91] Latombe, Jean-Claude, "Robot Motion Planning", *Kluwer Academic Publishers*, 1991.

[Koller93] Koller D., Daniilidis, K. and Nagel, H., "Model-based object tracking in monocular image sequences of road traffic scenes", *Int. J. Comput. Vision*, 10(3):257-281, 1993

[Ravela92] Ravela S., "A Survey of Reactivity", *COINS TR92-61*, Dept. of Computer Science, Univ. of Massachusetts, Amherst, 1992.

[Ravela94] Ravela S. and Weiss R., "Visual Servoing by Tracking Rigid 3D Motion",*To appear as Technical Report Nov. 1994*, Dept. of Computer Science, Univ. of Massachusetts, Amherst, 1994.

[Schultz94] Schultz, H., "Terrain Reconstruction from Oblique Views", *To appear in Proc. ARPA Image Understanding Workshop*, 1994.

[Shi94] Shi, J. and Tomasi, C., "Good features to track",*Proc. Comput. Vision Patt. Recognition*, pp. 593-600, 1994.

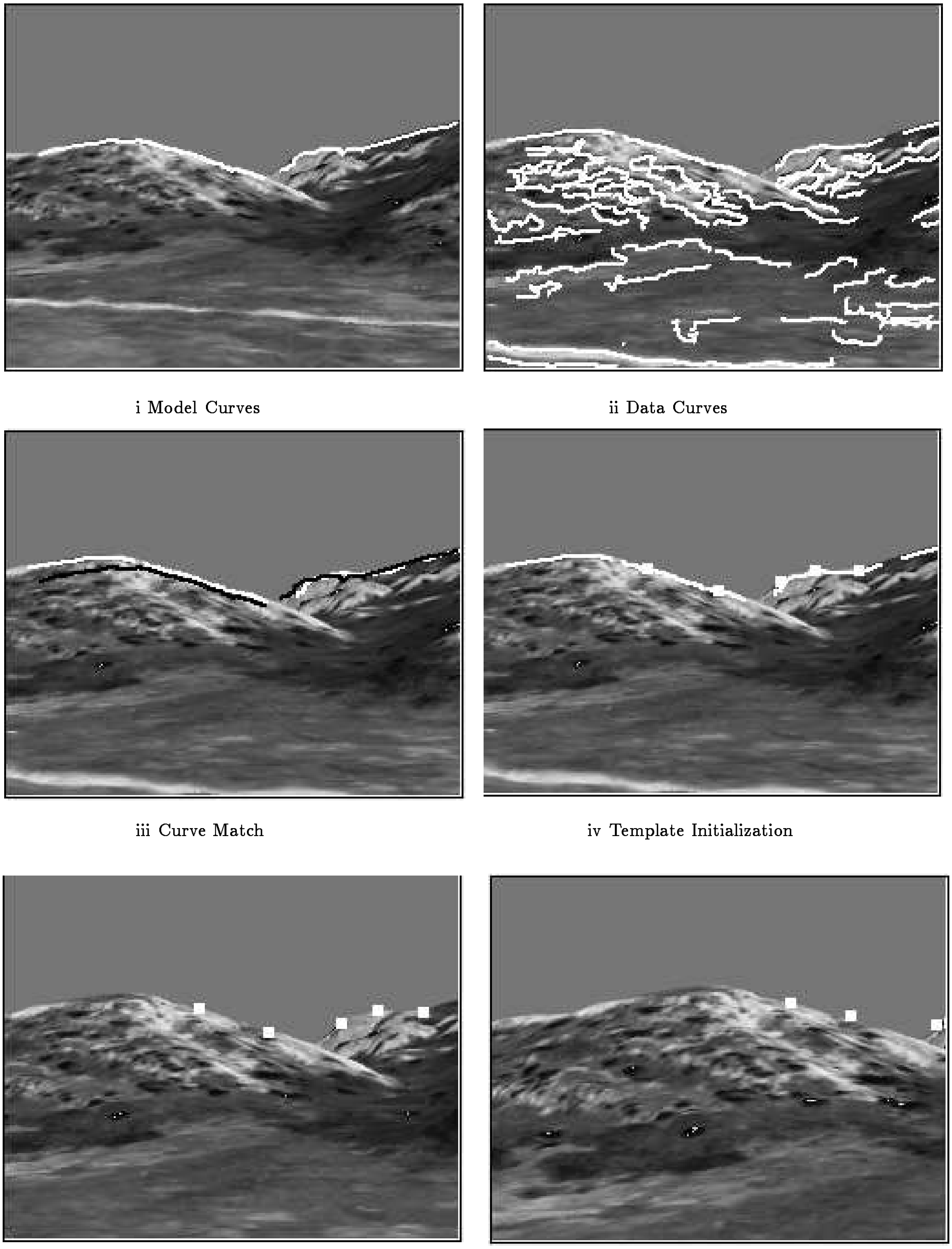
[Stan94] Stan, M. R., Burleson, W. P., Connolly, C. I. and Grupen, R. A., "Analog VLSI for Robot Path Planning", *J. VLSI Signal Processing*, 8(1):61-73, 1994.

[Tarassenko91] Tarassenko, L., and Blake, A., "Analogue Computation of Collision-Free Paths", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2102-2106, May 1990.

[Verghese90] Verghese, G., Gale, K. L., and Dyer, C. R.,"Real-time motion tracking of three dimensional objects", *Proc. IEEE conf. Robotics and Automation*, pp. 1998-2003, 1990.

[Wood83] Wood, G. A., "Realities with automatic correlation problem", *Photogram. Eng. and Rem. Sens.*, 49:537-538, 1993.

[Zhang93] Zhang, Z., Weiss, R., and Hanson, A., "Automatic Calibration and Visual Servoing For a Robot Navigation System", *CMPSCI TR93-14*, Dept. of Computer Science, Univ. of Massachusetts, Amherst, 1993.



i Model Curves

ii Data Curves

iii Curve Match

iv Template Initialization

v Tracking vi

Figure 4: **Matching and tracking horizon curves.**