

Dynamic Environment Exploration Using a Virtual White Cane

D. Yuan and R. Manduchi

Department of Computer Engineering, University of California, Santa Cruz

Abstract

The virtual white cane is a range sensing device based on active triangulation, that can measure distances at a rate of 15 measurements/second. A blind person can use this device for sensing the environment, pointing it as if it was a flashlight. Beside measuring distances, this device can detect surface discontinuities, such as the foot of a wall, a step, or a drop-off. This is obtained by analyzing the range data collected as the user swings the device around, tracking planar patches and finding discontinuities. In this paper we briefly describe the range sensing device, and present an online surface tracking algorithm, based on a Jump-Markov model. We show experimental results proving the robustness of the tracking system in real-world conditions.

1. Introduction

There are an estimated 200,000 totally blind individuals in the US [3]. Approximately five times this number are legally blind, but they still have some usable vision left which is not correctable by standard eyeglasses. They experience difficulty performing visual tasks, because of reduced acuity or field of view

Losing one's vision can be devastating for a number of reasons. In particular, one's ability to move about independently may be seriously affected. It has been estimated that more than 30% of the blind population do not ambulate autonomously outdoors [12]. Given the importance of mobility in everyone's daily life, education and technology that enable visually impaired persons to walk with confidence even in unfamiliar environments have a very significant social impact.

Different types of Electronic Travel Aids (ETA) have been developed over the past thirty years. Commercially available ETA's (which are meant only as secondary aid) include: the Mowat [19] and Sendero Group's MiniGuide [15], simple hand-held ultrasound devices that can be pointed toward any direction to give "clear path" information (a similar system mounted on a wheelchair is distributed under the name of Wheelchair Pathfinder [11]); the Nurion Laser Cane [9], embedded in a long cane to help detecting drop-offs and overhead obstacles; and the Sonic Pathfinder [10], an ultrasonic sonar device mounted on a

headband, which gives information about the surfaces in the immediate surroundings in the form of notes on a major musical scale¹.

This paper describes our recent research aimed at a novel ETA concept, targeting the category of visually impaired individuals unable or unwilling to fully rely on the long cane as a mobility device. The proposed ETA ("Virtual White Cane") is a hand-held instrument that measures distances to surfaces, using an active triangulation sensor composed by a laser pointer and a camera. Together with the range sensor, this device will have an onboard processor to analyze the range profiles acquired as the user scans the scene, and to detect features of interest such as curbs, corners, drop-offs, and stairways. Thus, the user will be able to explore the environment with a very narrow laser beam receiving information about features and their distances, and integrating it into a mental image of the scene. Note that laser technology allows for high spatial and temporal resolution that are far superior to those of existing ETA's based on ultrasound technology².

In order to validate our ideas, we have built a simple proof-of-concept prototype [18], shown in Figure 1 and briefly described in Section 2. This device produces range measurements at 15 Hz. This relatively high measurement rate can be used to track surfaces as the user moves the system around, pointing it at different areas in the scene. Surface tracking can provide precious information in the form of environmental features, which often presents themselves as geometric singularities.

In this paper we present a novel approach to surface tracking and geometric feature detection using our hand-held range sensing device. We use a Jump-Markov process to describe the evolution of the range data acquired by the device as it is pivoted in front of different types of surfaces. Our technique can reliably detect and classify different types of geometric singularities in the scene, which are important for a blind user's safe deambulation.

¹A similar device, the SonicGuide [13], mounted on a pair of spectacle frames, is currently out of production.

²The Nurion Laser Cane uses laser triangulation as well [9]. However, since the Laser Cane's purpose is not to compute distances, it only has a very limited amount of photoreceptors and no range processing capabilities. In addition, since it is embedded in a physical long cane, it cannot be used for scene exploration with the same freedom of movement as the proposed device.

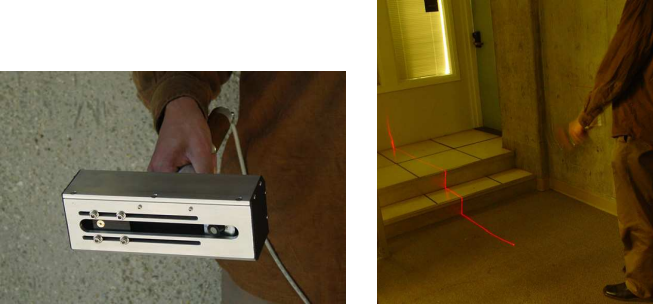


Figure 1: Left: The handheld device for range sensing. Through the opening slit, the laser (left) and the camera (right) can be seen. Right: The trace generated by the laser beam as the user pivots the device in an upward motion around a horizontal axis in front of two steps. (Note: this picture was taken with an exposure time of about 2 seconds.)

This paper is organized as follows. Sec. 2 describes our hand-held range sensing device. Sec. 3 discusses our model for the dynamics of range readings in front of piecewise planar surfaces, based on a Jump-Markov process formalism. A particle filter solution to the MAP feature detection is also presented. Sec. 4 presents quantitative experiments, while Sec.5 has the conclusions.

2. A Virtual White Cane

In order to make the paper self-contained, we provide here a brief description of our hand-held range sensing device, dubbed “Virtual White Cane”. For more details, please consult [18]. This sensor is based on active triangulation (see Fig. 2). Light is shone by a laser pointer (currently of Class 2, although we plan to migrate to a Class 1 pointer), and its reflection from a surface is detected by a miniaturized greyscale Firewire camera (PointGrey Dragonfly [16]), with a resolution of 1024×768 pixels, frame rate of 15 frames/s, and mounting a microlens of focal length $f = 8\text{mm}$. The laser has nominal divergence of 2 mrad and centerband at 650 nm. An optical filter, attached to the camera lens, reduces the effect of ambient light outside the laser’s bandwidth. The baseline B (that is, the distance of the camera to the laser) is equal to 80 mm, while the vergence β (the angle between the optical axis and the laser pointing direction) is equal to 11.1° . The law relating the position d of the light return on the camera (see Fig. 2) to the distance D to a surface is the following:

$$D = \frac{fB \cos \beta - dB \sin \beta + df}{d \cos \beta + f \sin \beta} \quad (1)$$

The determination of the epipolar line (that is, the line on the camera’s focal plane that is the locus of all the possible

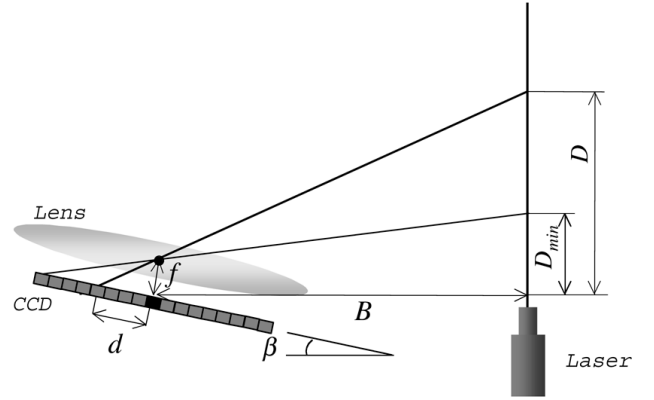


Figure 2: The layout of our laser triangulation system.

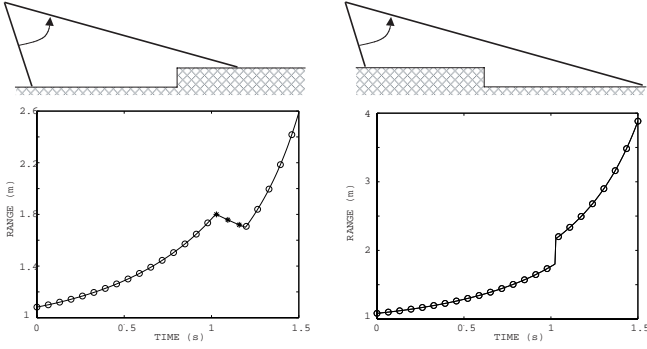
positions of the laser light return) is determined by a self-calibration procedure every time the device is turned on. The detection of the laser light return is performed using matched filtering. An automatic light compensation system controls the gain and the integration time of the camera. In our current prototype, a laptop computer connected to the camera is responsible for range sensing and processing.

Note that this system takes point measurements, which are immediately communicated to the user via acoustic or tactile feedback. A blind individual would thus scan the scene using our tool, relying on proprioception to determine the pointing direction and thus build a spatial map of the environment. An alternative would be to scan a whole area (for example, using a striper laser as in [14]), and then communicate important features (e.g., the presence of an obstacle) to the user.

3. Environment Sensing

A main goal of the proposed environment sensing module is to detect features that are important for safe deambulation. Obstacles that stick out from the ground at a certain height (say, 1 meter or higher) can be detected directly by aiming the range sensor at the space in front of the user. However, there are more insidious features such as steps, curbs, holes and drop-offs, which, if not detected, may be dangerous as the user may trip over them. These features are not detectable by point range measurement. They can, however, be detected by analyzing the time profile of the measured range as the user pivots the device around his/her wrist. For example, Fig. 3 shows the ideal time profiles of distance measurements in front of an ascending and of a descending step, when the device is rotated around an horizontal axis through its optical center in an upward and downward motion. The features corresponding to the base and edge

UPWARD MOTION



DOWNWARD MOTION

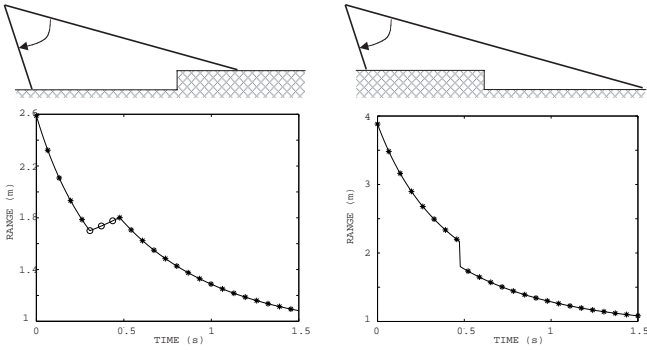


Figure 3: The time profiles of range as the device is pivoted around an horizontal axis in front of an ascending step (left column) and of a descending step or drop-off (right column). Segments of data belonging to different state evolution models are marked with different symbols (Model 1: circles; Model 2; stars).

of the ascending step and of the edge of the descending step are clearly recognizable from the time profiles of the range. Hence, it is conceivable that by modeling surfaces as piecewise planar, and by tracking the range using suitable models, discontinuities between surfaces can be detected and characterized. A similar problem have been studied for robotics navigation, with surfaces being tracked by a rotating lidar [2, 1]. However, whereas a lidar takes calibrated point measurements (i.e., the angular period between two consecutive measurements is known), the velocity at which the device is being pivoted, as well as the pivoting direction, are unknown to the processing module in our case. One possibility for taking direct angular measurements is to equip the device with a single-axis accelerometer, which can be used effectively as an inclinometer. This, however, would add to the hardware complexity of the system and therefore to its cost. In [18], the authors used an Extended Kalman

Filter to track planar surfaces under unknown (but constant) angular velocity. The dynamical model for the range, measured on a planar surface, was based on the following geometric identity, which relates the current measurement to the previous three measurements (see Fig. 4):

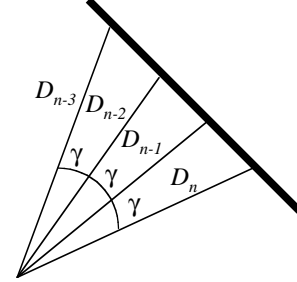


Figure 4: The measurement model for a planar surface, assuming constant angular velocity.

$$D_n = \frac{D_{n-1}}{\frac{D_{n-2}}{D_{n-3}} + \frac{D_{n-2}}{D_{n-1}} - \frac{D_{n-1}}{D_{n-2}}} \quad (2)$$

Points that are not well described by the planar surface model, and that are therefore likely to indicate a surface discontinuity, were identified based on a validation gate mechanism [8] (i.e., by thresholding the normalized posterior error variance). Although this strategy gave promising results, there are three major drawbacks to this approach. The first problem is that the validation gate mechanism is not very robust to other nuisances in the sensing process. For example, if the user varies the angular velocity of pivoting during a scan, this could mistakenly be interpreted as a discontinuity. In addition, it is difficult to characterize the type of a discontinuity (e.g., differentiate between a drop-off and a step).

Second, equation (2) requires that three previous consecutive measurements of the same surface type are available to predict the current measurement. Given the reading rate of the sensor (in our case, 15 readings per second), and the expected rotation rate (in our experience with user tests, this varies from 6°/s to 20°/s), this requirement translates into a possibly large minimum size of planar patches for successful tracking. For example, if the device is held at 1 meter of height, aimed at an area of the horizontal ground surface 2.5 meters ahead of the user, and undergoing an upward rotation of 10°/s, the distance between two consecutive reading points on the ground surface is of about 9 cm. Hence, a patch of at least 27 cm of horizontal surface before a discontinuity is needed for detecting the discontinuity in this situation. This can be a problem in the case of staircase tracking, when the size of the step is too small for reliable feature detection. A lower pivoting rate would reduce this

size requirement, but demands good control of the wrist motion to avoid vibration or changes in speed.

Third, the result of the validation gate procedure depends heavily on the chosen values for the state and observation variances, which requires some degree of ad-hoc tweaking. To overcome these drawbacks, we propose to use a different tracking scheme. First, the state dimension is reduced by only considering the past two measurements and using a locally linear model for state evolution. Second, a more sophisticated reasoning module is used for surface discontinuity detection, based on a Jump-Markov Process (JMP) structure. The higher robustness of the JMP approach compensates for the bias introduced by the suboptimal state evolution model. At the same time, simple dynamic modeling is needed for efficient implementation of the JMP tracker.

3.1. Jump-Markov Tracking – Overview

Jump-Markov Processes are a particular instance of Multiple Model processes. It is assumed that, at time n , the state evolves according to one of N possible stochastic models. The statistical description of the processes is known, but it is not known which process is effective at time n . In a Jump-Markov Process, the evolution in time of the model index is described by a Markov chain with known transition probabilities. Hence, JMP can be seen as a generalization of Hidden Markov Models (HMM), which are Markov chains seen in white noise. Even when each dynamic model in the JMP is linear with Gaussian noise, closed form recursive optimization results in the exponential explosion of candidate trajectories. JMP tracking is thus better performed by means of particle filtering [4, 5, 6, 17], which provides a compact description of the posterior state distribution, with complexity that remains constant through time. At each time period n , particles are sampled from the previously estimated posterior distribution and the current observation. Each particle is associated to one model and therefore one Kalman filter. The estimated state and model index at each time t can be computed based on the current posterior distribution of the state, represented by the particles and their weights.

3.2. Surface Models

Suppose the user is pivoting the sensor around an horizontal axis at constant angular rate in front of a surface. We'll approximate (2) with a second-order linear model:

$$D_n \approx D_{n-1} + (D_{n-1} - D_{n-2}) = 2D_{n-1} - D_{n-2} \quad (3)$$

Note that only two past samples are needed to predict the current sample. The bias introduced by this approximation depends on the distance to the surface, the angle at which the surface is seen, the angular rate, and the reading rate.

In order to detect surface discontinuities, it would be helpful to describe different surface orientations with different models. For example, one model would represent horizontal surfaces, while the other model would represent vertical surfaces. Unfortunately, since the direction of pivoting is unknown, it is impossible, based only on the analysis of the time profile of range, to differentiate between horizontal and vertical surfaces (see Fig. 3). It is, however, possible to discriminate between scan segments with increasing and decreasing measured distances. The first case corresponds to an horizontal surface seen by an upward scan or to a vertical surface seen by a downward scans. The second case corresponds to a vertical surface seen by an upward scan or to an horizontal surface seen by a downward scan³. In the first case we will say that the range evolves according to Model 1, and according to Model 2 in the second case. Indeed, the direction of switch between two models ($1 \rightarrow 2$ or $2 \rightarrow 1$) is sufficient to identify the type of surface discontinuity, regardless of the direction of rotation. As shown in Fig. 3, the first case ($1 \rightarrow 2$) occurs at the base of a wall or of a step, while the second case ($2 \rightarrow 1$) occurs at the upper edge of a step or of a curb. Being able to identify these two types of discontinuity would enable further, higher level reasoning about the scene. Unfortunately, the identity in (3) cannot be used to discriminate between Model 1 and Model 2. Hence, we propose to modify (3) into a model with two outcomes, one of which corresponds to Model 1 and the other one to Model 2:

$$\begin{aligned} \Delta_{n-1} &= D_{n-1} - D_{n-2} \\ D_n &\approx \begin{cases} \text{either } D_{n-1} + |\Delta_{n-1}| & (\text{Model 1}) \\ \text{or } D_{n-1} - |\Delta_{n-1}| & (\text{Model 2}) \end{cases} \end{aligned} \quad (4)$$

The two geometric models above are used to define two stochastic dynamic state equations for tracking distances. We will denote by $\mathbf{x}_n = [x_n(1), x_n(2)]'$ the state of the system at time n , with $x_n(1) = D_n$, $x_n(2) = D_{n-1}$. Then, the state evolution model is defined as follows:

$$\mathbf{x}_n = A\mathbf{x}_{n-1} + (-1)^{r_n} |B\mathbf{x}_{n-1}| + \mathbf{v}_n \quad (5)$$

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix} \quad (6)$$

where $r_n = 0$ if Model 1 applies, and $r_n = 1$ if Model 2 applies. \mathbf{v}_n is a zero-mean i.i.d. Gaussian noise with covariance $Q = \begin{pmatrix} \sigma_w^2 & 0 \\ 0 & 0 \end{pmatrix}$. The observation z_n is modeled as follows:

$$z_n = H\mathbf{x}_n + w_n, H = [1 \quad 0] \quad (7)$$

where w_n is a zero-mean i.i.d. Gaussian noise with variance σ_w^2 . We use the guidelines described in [18] for the choice of the noise variance in (5–7).

³We assume here that the user never points the laser above the horizon.

At first sight, state update equation (5) may seem redundant: after all, the simpler linear model $\mathbf{x}_n = \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{x}_{n-1} + \mathbf{v}_n$ can perfectly describe the geometric approximation in (3). However, by expressing the state evolution as in (5), we are able to impose prior constraints on the expected rate of transitions between different models (corresponding to features in the environments). More precisely, we model the evolution of the model index r_n as a Markov chain:

$$p(r_n | r_{1:n-1}) = M_{r_n, r_{n-1}} \quad (8)$$

where M is the transition probability matrix and $r_{1:n-1}$ represents the sequence of model indices until time $n-1$. Since model switches are relatively rare events, we usually set $M_{1,2} = M_{2,1} = 0.01$.

The goal of the feature detection system is to estimate, at each time n , the correct model index r_n , based on the sequence $z_{1:n}$ of observations. We will concentrate on MAP estimation of r_n , by determining the maximizer of the posterior mass distribution $p(r_n | z_{1:n})$. This operation should be performed in parallel with the estimation of the state \mathbf{x}_n .

A problem with the state evolution model (5) is that it is not linear in \mathbf{x} , which may impede direct use of Kalman filter for state prediction (for a fixed r_n). We can circumvent this problem by rewriting (5) as follows:

$$\begin{aligned} &\text{if } x_{n-1}[1] > x_{n-1}[2] && (9) \\ &\quad \text{if } r_n = 1 \textit{ (Model 1)} \\ &\quad \quad \mathbf{x}_n = C\mathbf{x}_{n-1} + \mathbf{v}_n, \\ &\quad \text{elseif } r_n = 2 \textit{ (Model 2)} \\ &\quad \quad \mathbf{x}_n = D\mathbf{x}_{n-1} + \mathbf{v}_n, \\ &\text{else} \\ &\quad \text{if } r_n = 1 \textit{ (Model 1)} \\ &\quad \quad \mathbf{x}_n = D\mathbf{x}_{n-1} + \mathbf{v}_n, \\ &\quad \text{elseif } r_n = 2 \textit{ (Model 2)} \\ &\quad \quad \mathbf{x}_n = C\mathbf{x}_{n-1} + \mathbf{v}_n, \end{aligned}$$

where

$$C = \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix}, D = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (10)$$

The state update equation (9) is linear, given the previous state \mathbf{x}_{n-1} and the current model index r_n . Unfortunately, states are never observed directly. We thus propose to substitute the terms in the first line of (9) with an approximation of \mathbf{x}_{n-1} . More precisely, rather than branching on the condition $x_{n-1}(1) > x_{n-1}(2)$, we will use either the condition $\hat{x}_{n-1}(1) > \hat{x}_{n-1}(2)$, where $\hat{\mathbf{x}}_{n-1}$ is the estimate of \mathbf{x}_{n-1} , or the condition $z_{n-1} > z_{n-2}$ on the observations.

3.3. Particle Filtering Implementation

Our algorithm for estimating the surface index r_n given the observations $z_{1:n}$ is based on particle filtering, as described in [5]. The idea is to represent the posterior distribution $p(r_n | z_{1:n})$ by a set of N particles⁴, $\{r_n^i\}$, with associated weights, w_n^i . Then, $p(r_n | z_{1:n})$ can be approximated by:

$$p(r_n | z_{1:n}) \approx \sum_i w_n^i \delta(r_n - r_n^i) \quad (11)$$

In other words, each particle votes with weight w_n^i to one value of r_n .

At each time period n , N new particles are sampled. More precisely, for i from 1 to N , a particle r_n^i is sampled from an importance density $\pi(r_n | z_{1:n}, r_{1:n-1}^i)$. Then, the importance weights of the particles are computed (up to a normalizing constant) as follows:

$$w_n^i \propto \frac{p(z_n | z_{1:n-1}, r_{1:n}^i) p(r_n^i | r_{n-1}^i)}{\pi(r_n^i | z_{1:n}, r_{1:n-1}^i)} w_{n-1}^i \quad (12)$$

and then all weights are normalized so that they sum up to one.

Given the simple form of our update equation, we are able to use the posterior $p(r_n | z_{1:n}, r_{1:n-1}^i)$ as importance density (the details are deferred to the Appendix). This choice is optimal in the sense that it minimizes the variance of the weights $\{w_n^i\}$, thereby maximizing the effective sample size [7]. This is seen by noticing that, for any given i , the weight w_n^i is proportional to $p(z_n | z_{1:n-1}, r_{1:n-1}^i)$, and therefore it is independent of the actual sample r_n^i . Being able to sample from $p(r_n | z_{1:n}, r_{1:n-1}^i)$ reduces the risk of algorithm degeneracy. Indeed, in our tests, we never found any occurrence of degeneracy, and therefore didn't need to implement resampling procedures.

3.4. Drop-Off Detection

As shown in Fig. 3, the surface model index never changes across a drop-off. Hence, the method outlined in the previous section cannot be applied here. In fact, we observed that a drop-off always determines a noticeable ‘‘jump’’ (discontinuity) in the tracked data. Accordingly, we propose to detect such discontinuities using a validation gate on the variance of the estimation error for the state \mathbf{x}_n , computed using the EKF tracker of [18] running in parallel to the JMP tracker. Whereas, as stressed earlier, the validation gate approach is not sufficiently reliable for the detection of model switches, our experience has shown that it can be a valid indicator of drop-off occurrences.

Due to its large error variance, the state \mathbf{x}_n is not reliably estimated by the system at the jump location and for a few samples afterwards. For this reason, in these situations the

⁴In our implementation, we set $N = 100$.

condition $x_{n-1}(1) > x_{n-1}(2)$ in (9) is approximated by the condition $z_{n-1} > z_{n-2}$. In all other cases, the condition is computed on the estimated state \hat{x}_{n-1} .

4. Experiments

We ran a number of different experiments with our feature detection algorithm, in different conditions of illumination and with different surface types. The algorithm runs in real time with the sensor’s reading rate of 15 frames/second. Fig. 5 and 6 show two examples of range acquisition and feature detection in front of an ascending and of descending staircase respectively. $1 \rightarrow 2$ model switches are represented by diamonds, while $2 \rightarrow 1$ model switches are represented by squares. Circles represent jumps (drop-offs). The range varied between 0.7 and 3 meters. Almost all features are correctly detected, with very few false positives.

In order to provide a quantitative assessment of the algorithm’s performances, we ran a number of tests with repeated pivoting at variable angular speed in front of the same ascending and descending staircases of Fig. 5 and 6. ROC curves for model switches and jump detection were measured on the ascending and on the descending staircase test respectively. In the first case, out of 1766 measurements, we manually identified 321 model switches, which represent the ground truth for comparison. In the second case, out of 753 measurements, there were 90 ground truth jumps. The ROC curves for the two cases, plotting sensitivity versus selectivity (the latter represented with an inverted scale as customary) are shown in Fig. 7. The ROC curve for model switch detection was built by varying the transition probability $M_{1,2} = M_{2,1}$ in (8), while in the jump detection case we varied the validation gate for the EKF model [18]. Note that model switch and jump detection are not independent processes (e.g., detecting a jump conditions the future model switch detection, and vice-versa). This is the reason for the non-monotonic behavior of jump model detection sensitivity as the validation gate is increased, which is noticeable in Fig. 7.

5. Conclusions

We have presented an algorithm for the automatic detection of geometric singularities as a blind user scans the scene with our hand-held range sensing device (Virtual White Cane). A Jump-Markov process was used to describe the evolution of range measurements in a piecewise planar world model. Quantitative experiments have been carried out to assess the performances of the geometric feature detection algorithm.

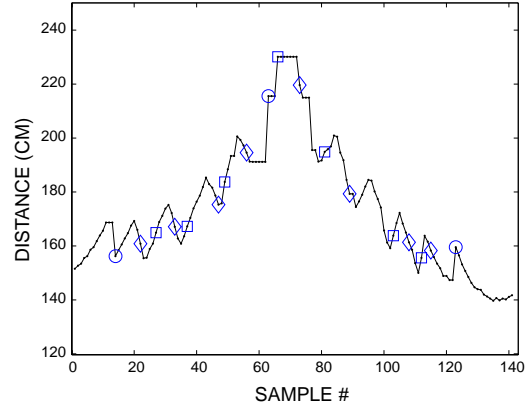


Figure 5: Experiments with an ascending staircase. Dots: actual measurements. Diamonds: $1 \rightarrow 2$ switches. Squares: $2 \rightarrow 1$ switches. Circles: jumps.

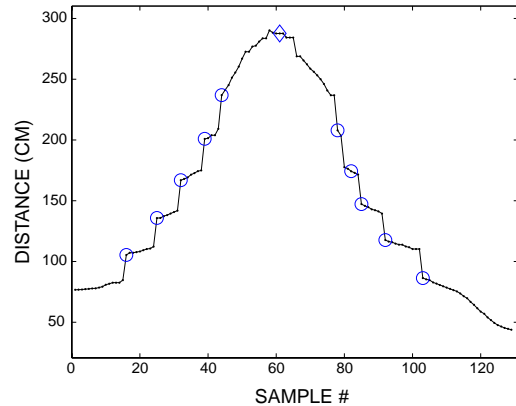


Figure 6: Experiments with a descending staircase. Dots: actual measurements. Diamonds: $1 \rightarrow 2$ switches. Circles: jumps.

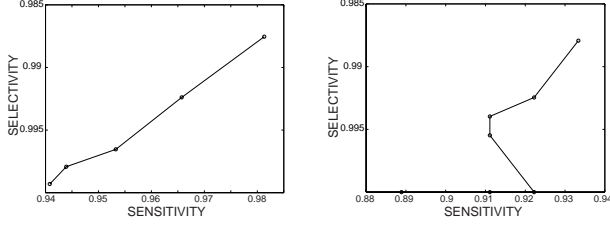


Figure 7: ROC curves for model switch detection (left) and jump detection (right).

Appendix

In this Appendix we show how, based on our dynamic model, it is possible to compute $p(r_n | z_{1:n}, r_{1:n-1}^i)$. Using Bayes' rule, one can write:

$$p(r_n | z_{1:n}, r_{1:n-1}^i) \propto p(z_n | z_{1:n-1}, r_{n-1}^i, r_n) p(r_n | z_{1:n-1}, r_{1:n-1}^i) \quad (13)$$

where $p(r_n | z_{1:n-1}, r_{1:n-1}^i) = p(r_n | r_{n-1}^i)$ according to the basic Markovian hypothesis. The first factor in (13) can be rewritten as

$$p(z_n | z_{1:n-1}, r_{n-1}^i, r_n) = \int p(z_n | z_{1:n-1}, r_{1:n-1}^i, r_n, \mathbf{x}_n) p(\mathbf{x}_n | z_{1:n-1}, r_{1:n-1}^i, r_n) d\mathbf{x}_n \quad (14)$$

The first factor in the integral is equal to $p(z_n | \mathbf{x}_n)$ (according to (7)), and therefore it is a Gaussian variable. The second factor represents the estimate of \mathbf{x}_n given the sequence of past model indices $r_{1:n-1}^i$, the model index r_n , and the previous observations $z_{1:n-1}$. Hence, this is again a Gaussian variable, with parameters that depend on r_n .

References

- [1] M.D. Adams. On-line gradient based surface discontinuity detection for outdoor scanning range sensors. In *Proc. 2001 IEEE IROS*, pages 1726–31, Maui, HI, 2001.
- [2] M.D. Adams and A. Kerstens. Tracking naturally occurring indoor features in 2-D and 3-D with lidar range/amplitude data. *The International Journal of Robotics Research*, pages 907–23, 1998.
- [3] American Foundation for the Blind. www.afb.org.
- [4] Arnaud Doucet, Andrew Logothetis and Vikram Krishnamurthy. Stochastic Sampling Algorithms for State Estimation of Jump Markov Linear Systems. *IEEE Transactions on Automatic Control*, 45(1):188–202, 2000.
- [5] Arnaud Doucet, Neil J. Gordon and Vikram Krishnamurthy. Particle Filter for State Estimation of Jump Markov Linear Systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, 2001.
- [6] Arnaud Doucet and Christophe Andrieu. Iterative Algorithms for State Estimation of Jump Markov Linear Systems. *IEEE Transactions on Signal Processing*, 49(6):1216–1227, 2001.
- [7] S. Arulampalam, A. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Trans. Signal Processing*, 50(2):174–88, Feb. 2002.
- [8] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
- [9] J.M. Benjamin, N.A. Ali, and A.F. Schepsis. A laser cane for the blind. In *Proc. San Diego Biomedical Symposium*, volume 12, pages 53–7, 1973.
- [10] D. Bissitt and A.D. Heyes. An application of biofeedback in the rehabilitation of the blind. *Applied Ergonomics*, 11(1):31–3, 1980.
- [11] B.B. Blasch, W.R. Wiener, and R.L. Welsh. *Foundations of Orientation and Mobility*. AFB Press, 1997. Second Edition.
- [12] D.D. Clark-Carter, A.D. Heyes, and C.I. Howarth. The effect of non-visual preview upon the walking speed of visually impaired people. *Ergonomics*, 29(12):1575–81, 1986.
- [13] L. Kay. A sonar aid to enhance spatial perception of the blind: Engineering design and evaluation. *Radio and Electronic Engineer*, 44(1):605–27, 1974.
- [14] C. Mertz, J. Kozar, J.R. Miller, and C. Thorpe. Eye-safe laser line striper for outside use. In *IV 2002, IEEE Intelligent Vehicle Symposium*, 2002.
- [15] The Miniguide ultrasonic mobility aid. www.gdp-research.com.au/ultra.htm.
- [16] Inc. Point Grey Research. www.ptgrey.com/.
- [17] V. P. Jilkov and X. R. Li. Online Bayesian Estimation of Transition Probabilities for Markovian Jump Systems. *IEEE Transactions on Signal Processing*, 52(6), 2004.
- [18] D. Yuan and R. Manduchi. A Tool for Range Sensing and Environment Discovery for the Blind. *IEEE Workshop on Real Time 3D Sensor and Their Use*, Washington, DC, June 2004.
- [19] Wormald International Sensory Aids, 6140 Horseshoe Bar Rd., Loomis, CA 95650.