TÜBİTAK

Turk J Elec Eng & Comp Sci, Vol.19, No.1, 2011, © TÜBİTAK
doi:10.3906/elk-0910-272

Line detection with adaptive random samples

Ali Cafer GÜRBÜZ
Department of Electric and Electronics Engineering, TOBB University of Economics and Technology, Sögütözü Cad. No 43, Ankara-TURKEY
e-mail: acgurbuz@etu.edu.tr

Abstract

This paper examines the detection of parameterized shapes in multidimensional noisy grayscale images. A novel shape detection algorithm utilizing random sample theory is presented. Although the method can be generalized, line detection is detailed. Each line in the image corresponds to a point in the line parameter space. The method creates hypothesis lines by randomly selecting parameter space points and tests the surrounding regions for acceptable linear features. The information obtained from each randomly selected line is used to update the parameter distribution, which reduces the required number of random trials. The selected lines are re-estimated within a smaller search space with a more accurate algorithm like the Hough transform (HT). Faster results are obtained compared to HT, without losing performance as in other faster HT variants. The method is robust and suitable for binary or grayscale images. Results are given from both simulated and experimental subsurface seismic and ground penetrating radar (GPR) images when searching for features like pipes or tunnels.

Key Words: *Line detection, Hough Transform, Tunnel Detection, Random sampling, Subsurface shape detection*

1. Introduction

Feature detection is the extraction of necessary information from an image by means of signal processing tools. It is an important and broad topic extensively studied in areas like image processing [1, 2, 3, 4], computer vision [5, 6, 7] and subsurface imaging [8, 9]. The Hough Transform (HT) [10], its variants and their generalizations [11, 12, 13, 2] are the most commonly used methods capable of detecting features such as lines[12, 6], circles[13], or any other parameterized curve[14, 15]. The HT uses a parameterized model of each feature to transform the feature in the original image space into a single mesh point in the parameter space. The better a feature corresponds to the model, the more values (votes) will be accumulated at a mesh point. Features having votes above a predetermined threshold are selected as a detection. Although the HT is effective even for very noisy images, it is not easily implementable because of its high computation time and large memory requirements. These problems are exacerbated as the dimension of the search space increases.

Various methods have been proposed to decrease the computation requirements of the HT. The primary ones including randomness are the Probabilistic Hough Transform (PHT) [16], the Randomized Hough Transform (RHT) [17, 18], and Line Detection using Random Sample Consensus (RANSAC) [19, 20, 21]. The PHT uses only a randomly selected subset of edge points in the image as input for the HT; however, this technique leads to erroneous results for a small subset, and selection of the optimum subset size requires knowledge of the image. Although the latter problem is solved by the Progressive PHT (PPHT)[22] algorithm, its performance is still worse than the HT.

The RHT randomly selects n pixels, solves for the feature parameters and then increases the value of the parameter space cell by one. Using a many-to-one mapping and randomization saves on memory requirements as well as computation time. The RHT is suitable for low noise images [23]. For highly noisy images, the PHT outperforms the RHT, but neither algorithm works very well in detecting features in extremely noisy images.

The methods based on hierarchical division and pruning of the parameter space, like the fast HT (FHT) [24] or the Adaptive HT [25], recursively divide the parameter space into hypercubes and perform the HT only on the hypercubes with votes exceeding a selected threshold. More robust methods [26, 27] that propagate the error to the parameter space have also been developed. Although the existing hierarchical HT methods succeed in decreasing the computational load of the HT, they were only applied to binary edge images.

To improve the robustness of feature detection, the RANSAC [19] algorithm was proposed. In RANSAC n edge points are randomly selected, and data lying within a defined distance from the line are classified as *inliers* with the remaining data marked as *outliers.* If the number of inliers is larger than a certain threshold, the feature parameters are re-estimated using only the inliers. In this way, the effect of misleading outliers is mitigated. Furthermore, the memory requirements are much less for the RANSAC algorithm since an accumulator array is not utilized. RANSAC has been shown to perform line detection faster than the HT [20]. Small improvements for increasing RANSAC's performance were proposed in [21, 28] for the line detection problem.

A variant of the RANSAC algorithm specific to line detection [21] increases the probability of selecting edge pixels on the true line by randomly choosing three points, instead of just two, an applying a distance criteria to estimate the edge. The efficiency of the RANSAC algorithm was improved by [28] by determining the inliers without performing an explicit distance calculation.

Prior research has established an extensive basis for faster and more robust feature detection. However, some of the existing methods can only be applied to binary images. The methods applicable to grayscale images suffer degraded performance compared to the HT. Our goal is to detect features in highly noisy grayscale images robustly and faster than the HT without sacrificing performance. Our motivation is detecting buried linear structures, such as tunnels or pipes, in subsurface images generated from ground penetrating radar (GPR) or seismic measurements. This is a problem of great interest in industrial/civil engineering, as well as in military applications [9, 8, 29, 30, 31]. Typically, subsurface structures are visually masked in these images by noise and clutter, rendering their detection impossible without the implementation of feature detection algorithms. Quantizing the subsurface images by binarization to accommodate existing methods using edge detection may seem like a viable solution, but the loss of important data degrades performance, especially in high noise images where it is not even possible to create edges when the features are masked by the noise.

The proposed adaptive random sample theory (ARST) method creates random hypothesis features and tests not only the selected feature, but also the region around it, to look for acceptable shapes. Contrary to choosing n random points to define a feature and solving for its parameters, the feature parameters are

randomly selected from a parameter space distribution. The information obtained from each randomly selected feature is used to update this distribution, which reduces the total required number of random trials. The selected features are re-estimated within a smaller search space with a more accurate algorithm like the HT. Results show that this two-stage algorithm decreases the total computation time by limiting the search space over which the HT is performed to smaller sets. The performance is not degraded because the small sets are randomly chosen probable areas. The proposed algorithm is described in Section 2. Simulated and experimental results from subsurface seismic and GPR images are given in Section 3.

2. Proposed algorithm

The basic idea of the adaptive random sample theory (ARST) algorithm is to first find rough areas or volumes in the image that possibly include features and then search only these rough regions with a more accurate algorithm like the HT. Reducing the search space of the HT decreases the computation time, while still maintaining detection performance at a level comparable to that of the full HT.

The features to be detected are a subset of the parameter set $\mathcal{P} = \{ p_1, p_2, \ldots, p_n \}$, where each parameter p_i has limits defined by the sets $\mathcal{R}_i$. For example, a line in 2D is denoted by two parameters, (ρ, θ), as $\rho = x\cos\theta + y\sin\theta$ and the ranges of these parameters are

$$\begin{aligned} \mathcal{R}_\rho &= \{\rho : |\rho| < r_{\max}\} \\ \mathcal{R}_\theta &= \{\theta : 0 < \theta < \pi\} \end{aligned} \tag{1}$$

where $r_{\max} = \sqrt{s_x^2 + s_y^2}$ for an $s_x \times s_y$ image. Line parametrization in 3D would require four parameters [15].

2.1. Algorithm steps

The ARST method has two stages: the first stage searches for approximate candidate features, while the second stage refines the estimate of the features. A step by step description of the algorithm follows.

Stage I (Candidate Model Selection)

(i) Generate a candidate feature by randomly sampling the current parameter distribution F, i.e.,

$$(p_1, p_2, \ldots, p_n) \sim F^k$$

during iteration k. The parameter distribution F can be initialized as uniform if there is no *a priori* information. The randomly selected feature vector $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)$ instantiates one possible feature in the image.

In RANSAC and RHT, random features are computed from n randomly chosen points in the image While this might be time efficient for binary images where edge points do not constitute a majority of the image, for gray-scale and higher dimensional images, this requires a significant amount of computation. Our method avoids these inefficiencies by drawing samples directly from the parameter distribution.

(ii) Calculate a feature indication metric C_S using the image data within a distance σ of the selected feature. For subsurface images the high reflected power from the subsurface targets is an indication of target presence, thus C_S is calculated by summing up the pixel values for the defined region. In this way, the algorithm is

made robust against discretization errors or measurement errors within the region. For binary images C_S is the number of edge points within the distance σ, analogous to RANSAC's step in which the number of inliers within a distance of the selected line is counted. When extracting road networks in SAR images, C_S can be calculated as in [32] to find where the road is darker than its neighbors.

Figure 1(a) shows the strip summation region surrounding the centerline with parameters (ρ, θ). All the lines must have parameters $(\rho \pm \Delta\rho, \theta \pm \Delta\theta)$ that are within the parameter region shown in Figure 1(b) and such that

$$\Delta\rho = \sigma^* \quad \text{and} \quad \Delta\theta = \arctan\left\{\frac{2(\sigma - \sigma^*)}{r_{\max}}\right\} \tag{2}$$

where σ^* is shown in Figure 1(b).

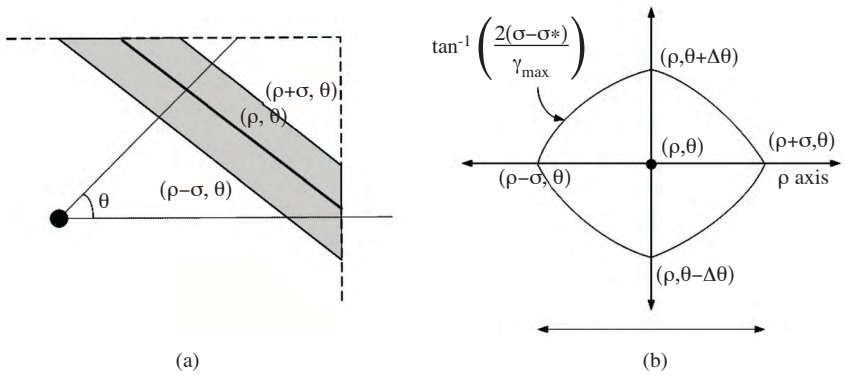

Figure 1. (a) Randomly selected strip region; and (b) the parameter space corresponding to the strip region in (a).

For small $\Delta\theta$, the parameter domain shape for the strip region becomes a diamond, where $\Delta\theta = \frac{2\sigma}{r_{\max}}$ and $\Delta\rho = \sigma$. The small $\Delta\theta$ approximation also puts an upper bound on the σ parameter, e.g., $\Delta\theta < \pi/6$ gives $\sigma_{\max} \cong r_{\max}/4$. Selection of the σ parameter affects the run time of the algorithm, and its optimal selection is discussed in Section 3.1.

(iii) If the feature indication metric C_S from Step (ii) is less than a preset threshold T_s (see (5) below) then we conclude that none of the features in the parameter space corresponding to the randomly selected strip region can be accepted, and we skip to the parameter space update in Step (iv). If $C_S \geq T_s$ then the randomly selected (ρ, θ) is added to the candidate feature list (CFL) before updating the parameter space in Step (iv).

(iv) Update the parameter distribution as

$$F^{k+1} = F^k$$

$$F^{k+1}(p_1 - \Delta p_1 : p_1 + \Delta p_1, \ldots, p_n - \Delta p_n : p_n + \Delta p_n) = 0$$

$$F^{k+1} = \frac{F^{k+1}}{\sum F^{k+1}},$$

where F^k is the parameter distribution at the k^{th} iteration.

(v) After updating the parameter distribution, increment k. If $k < k_{\max}$ go to Step (ii); otherwise, terminate the iterations.

Stage II (Refine The Estimate) For each candidate feature $\boldsymbol{p}_i$ in the CFL, we define a search region $\mathcal{S}_i$ surrounding $\boldsymbol{p}_i$ in the parameter space, and we apply the full HT over this region to find better estimates of the features in that region. For example, the search region $\mathcal{S}_i$ for the 2D case would be the approximate diamond in Figure 1(b).

This stage also enables multiple features within a region to be resolved and successfully detected. For example, if two lines are close to each other Stage I selects only one random feature in that region. To the extent that the HT can resolve multiple features, all of them can be detected in Stage II.

2.2. Selection of algorithm parameters

Selection of the parameter σ is important because it determines the number of trials $k_{\max}$ in Stage I, and size of the search space, $\{\Delta p_{1_i}, \ldots, \Delta p_{n_i}\}$, in Stage II. The relative size of σ with respect to the size of the image, $r_{\max}$, is a crucial parameter. Selecting a very small $\sigma/r_{\max}$ ratio will reduce the size of $\mathcal{S}_i$, but increase $k_{\max}$. In the limit $\sigma \to 0$, the algorithm approaches the RHT. Conversely, when the ratio $\sigma/r_{\max}$ increases, the size of $\mathcal{S}_i$ will increase, while the value of $k_{\max}$ will decrease. In the limit $\sigma \to r_{\max}$, the number of trials in Stage I will be 1, making the candidate model selection stage useless and reducing the proposed method to the HT applied in Stage II to the whole image. So ARST is a new method standing between RHT and HT.

Once $\sigma/r_{\max}$ is selected, the Δp_{ℓ_i} can be computed as

$$\begin{array}{ll} \underline{\text{2D Line Detection}} & \underline{\text{3D Line Detection}} \\ \Delta\theta = \arctan\left\{\dfrac{2\sigma}{r_{\max}}\right\} & \Delta\theta = \arctan\left\{\dfrac{2\sigma}{r_{\max}}\right\} \\ \Delta\rho = \sigma & \Delta\varphi = \arctan\left\{\dfrac{2\sigma}{r_{\max}}\right\} \\ & \Delta u = \Delta v = \sigma \end{array} \tag{3}$$

for detecting lines in 2D and 3D[1].

Let $k_{\max}$ be the number of trials required to have at least one feature within the $\mathcal{S}_i$ vicinity of the i^{th} true feature parameter with probability q. If the randomly selected parameters fall outside of $\mathcal{S}_i$, the feature cannot be correctly detected since Stage II only searches for features within $\mathcal{S}_i$.

In the adaptive selection of parameters, at each iteration part of the distribution of size $(2\Delta p_1)(2\Delta p_2)\cdots(2\Delta p_n)$ is removed. The probability $1-q$, which is equal to not getting any random selection in the $\mathcal{S}_i$ vicinities of the true feature parameter during any of the $k_{\max}$ trials, can be reduced to

$$1 - q = \frac{\prod_{\ell=1}^{n} |\mathcal{R}_\ell| - k_{\max} 2^n \prod_{\ell=1}^{n} \Delta p_\ell}{\prod_{\ell=1}^{n} |\mathcal{R}_\ell|}. \tag{4}$$

[1]The 3D line parametrization in [15] is used.

In (4), n denotes the number of parameters representing the feature, the denominator is the total volume of the n dimensional parameter space and the numerator is the left parameter space after $k_{\max}$ removals. Thus $n = 2$ for a line in 2D and $n = 4$ for a line in 3D. As an example, consider a 2D image of size 100×100. Using a resolution of 1 in ρ and 1° in θ results in 50912 line summations for HT. If $\sigma = 10$ is selected, then from (3) $\Delta\rho = 10$ and $\Delta\theta = 8^\circ$. A detection probability of $q = 0.99$ requires a minimum of only 157 strip summations for Stage I, while Stage II requires to make 320 line summations for the selected $\mathcal{S}_i$. In total the computational load of HT is decreased and results indicating this are given in section 3.1 and section 3.2.

Selection of the optimal threshold T_s is a hard problem. Depending on the image type, the accepted feature criteria and knowledge of the image statistics, a suboptimal T_s can be selected. For the problem of feature detection in subsurface images, the noise statistics (μ_ν, σ_ν^2) of the image would have to be estimated from the image itself. Feature detection of an SNR level greater than m_{SNR} would use a threshold T_s that should be selected as

$$T_s = \left(m_{\text{SNR}}\sigma_\nu^2\right) s_i + N_{\text{pix}}\mu_\nu, \tag{5}$$

where $\left(m_{\text{SNR}}\sigma_\nu^2\right)$ is the signal power, and N_{pix} is the number of pixels added in the summation region of the selected feature. With this selection if there is a line with average SNR value of at least m_{SNR} the total summation from the region should be greater than T_s.

3. Results

To illustrate how the proposed algorithm works, a 101×101 image is created with two linear structures having the parameter values $\rho_1 = 70$, $\theta_1 = 65^\circ$ and $\rho_2 = 20$, $\theta_2 = 120^\circ$; the two lines are shown in Figure 2(a). Zero-mean white Gaussian noise is added to the image with an SNR of 0 dB. Figure 2(b) shows the synthetic linear structures with noise added. The linear features cannot be seen and are masked by the noise.

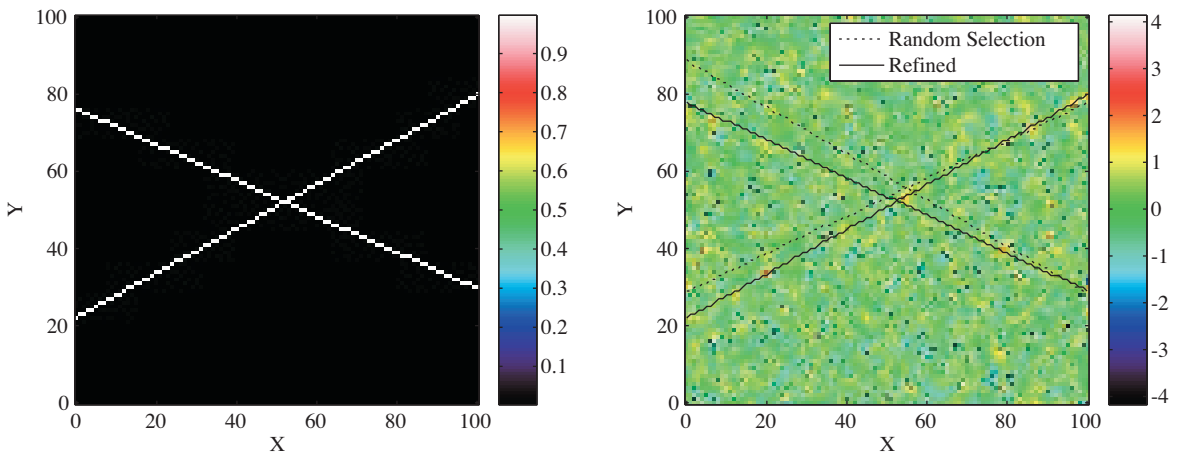

Figure 2. (a) A 101×101 image containing two linear features. (b) The image with lines detected from Stage I (random selection) and Stage II (refined by the HT) of the proposed algorithm on the noisy image of SNR 0 dB.

The distance parameter σ is chosen as 10. The number of random trials, $k_{\max}$, is calculated using (4). The candidate features obtained from Stage I of the algorithm are shown in Figure 2(b) with dashed lines. These

candidate features are refined in Stage II by applying the HT within a vicinity $(\pm\Delta\rho, \pm\Delta\theta)$ of the parameters of the selected lines as pictured in Figure 1(b). The refined lines are shown by solid lines in Figure 2(b). The true parameters along with the estimates obtained from both stages of the algorithm are listed in Table 1.

Table 1. True and detected target parameters

2D Line Detection Results				
Targets	Target 1		Target 2	
Parameters	ρ	$\theta(^\circ)$	ρ	$\theta(^\circ)$
True Parameters	70	65	20	120
Random Selection (Stage I)	76	59	27	116
Refined by HT (Stage II)	70	64	19	120

3.1. Performance of the algorithm for different σ values

The parameter σ, discussed in Section 2.2, determines the algorithm parameters $\{\Delta p_{1_i}, \ldots, \Delta p_{n_i}\}$ and $k_{\max}$. In this section, we analyze the effect of σ on the run time of the algorithm using a Monte-Carlo simulation and select the σ that minimizes the total run time. A 2D image of size 100×100 containing one linear structure with parameters $(\rho, \theta) = (70, 65^\circ)$ with SNR $= 10\,\text{dB}$ is generated.

For various σ values, the algorithm was run 100 times and the elapsed times for Stages I and II were recorded. Increasing σ will decrease the number of trials by (4) for finding rough target areas. Time for Stage I of the algorithm decreases inversely with σ since time spent in Stage I is only related with how many number of trials are done. In the limit where σ/s_i is close to 0.5 which means the whole image is considered at once, the number of trials is zero, thus no time is spent in Stage I. This whole effect can also be observed in Figure 3(a) where the average run time for Stage I of the algorithm was obtained from the simulations.

The parameter space area defined in (3) searched by the HT in Stage II of the algorithm increases quadratically with σ for the 2D case. This is because the area searched by the HT increases with increasing σ. A quadratically increasing run time is also observed in Figure 3(b) with the simulations as expected. This quadratic increase is for line detection in 2D images, but for 3D images the search space size increases as σ^4.

When σ is small number of trials needed to find rough areas increases but since the area for HT is small time in Stage II is less. Increasing σ decreases the number of trails in Stage I, thus the time spent in that stage but since the search area for HT is increased Stage II time increases with σ. The combination of these two effects yields the total average run time shown in Figure 3(c). For the total run time of the algorithm, a σ/s_i value that minimizes the average run time can be found, and $\sigma/s_i = 0.1$ is selected as the optimum parameter for line detection in 2D images.

3.2. Performance of the proposed method for varying SNR

The detection performance and average running time of the ARST algorithm for varying SNR levels are now compared to the HT, RHT and PHT for 2D images. Two versions of the RHT and PHT algorithms are tested: (RHTL, PHTL) and (RHTH, PHTH). The PHT uses 50% and 5% of the data for the PHTH and PHTL results; RHT uses 10^5 and 10^4 random point selections for RHTH and RHTL, respectively. At each SNR

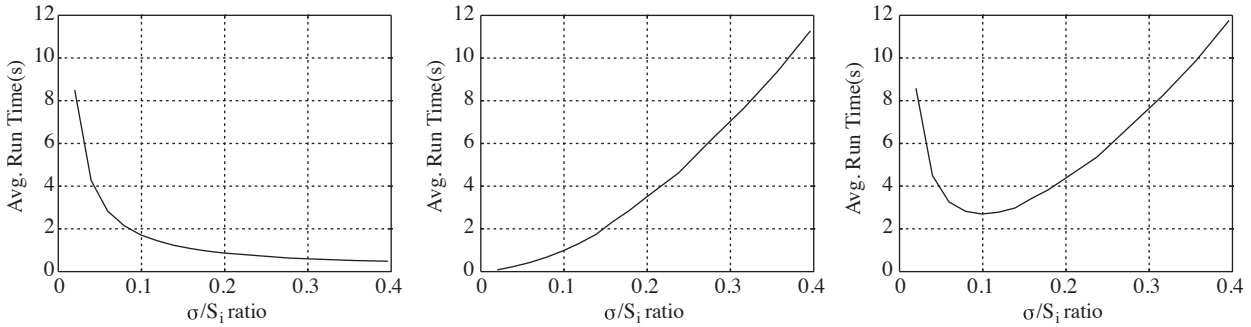

Figure 3. (a) Average run time of stage I of the algorithm versus σ/s_i, (b) Average run time of stage II of the algorithm, and (c) Total average run time of the algorithm. The size of the image is $s_i = 100$ for this simulation.

value, all six algorithms are run 500 times with random noise added to the original signal each time. The detected target parameters and the run times of the algorithms are noted. For a fair comparison among all six algorithms, identical parameter resolutions are used, i.e., 2° in θ and 1 in ρ. The ARST algorithm uses a ratio of $\sigma/s_i = 0.1$.

The probability of detection (P_D) for all six algorithms is plotted in Figure 4 which shows that the ARST and the HT have nearly the same performance for all SNR values while the PHT and RHT algorithms have much lower P_D. The advantage of the ARST lies in its lower average run time. The average run times of the algorithms are given in Table 2. It can be seen that while the ARST has the same performance level as the HT, the average run time is nearly ten times less than the HT. The RHTL and PHTL algorithms can run faster than the proposed ARST method, but they have much worse detection performance. Even the PHTH and RHTH algorithms, which have higher average run times than ARST, have worse detection performance. So the proposed method combines the fast running time of a random selection method with the best possible detection performance of the HT.

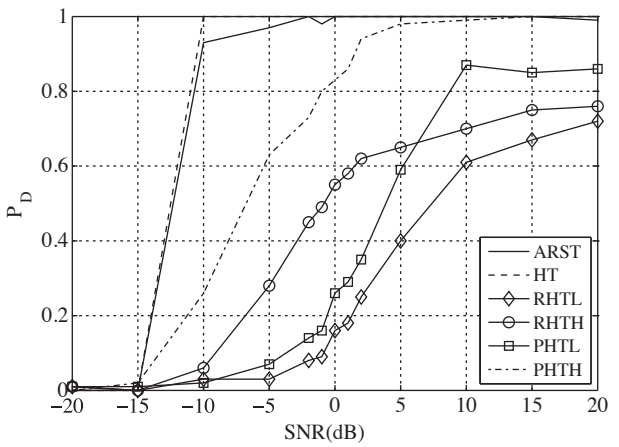

Figure 4. Probability of detection P_D vs. SNR for the six different feature detection algorithms.

Table 2. Average run times (s) of the algorithms in 2D

ARST	HT	RHTH	RHTL	PHTH	PHTL
1.72	16.13	4.47	0.44	7.75	0.77

3.3. Experimental data results in 3D

To investigate the potential of new feature detection algorithms an experimental setup using co-located GPR and seismic sensors was built [29, 33, 34]. Two different experimental scenarios have been investigated with scale models of linear structures in an experimental model sandbox filled with nearly homogeneous sand as shown in Figure 5(a). In Scenario 1 a scale model for a tunnel is buried within a 1.8 m × 1.8 m region in the center of the tank. The tunnel is 10 cm in diameter and is buried approximately 58 cm deep (the depth varies from 53 to 63 cm). This diameter makes a 20 to 1 scale model for a shallow tunnel just big enough for a man to slide through. The sensors are scanned over this region with a computer controlled positioner. To investigate a second, more difficult, configuration, three PVC pipes with diameters of 1.27 cm, 2.54 cm, and 5.08 cm were buried with variable depths as shown in Figure 5(b) for the second scenario.

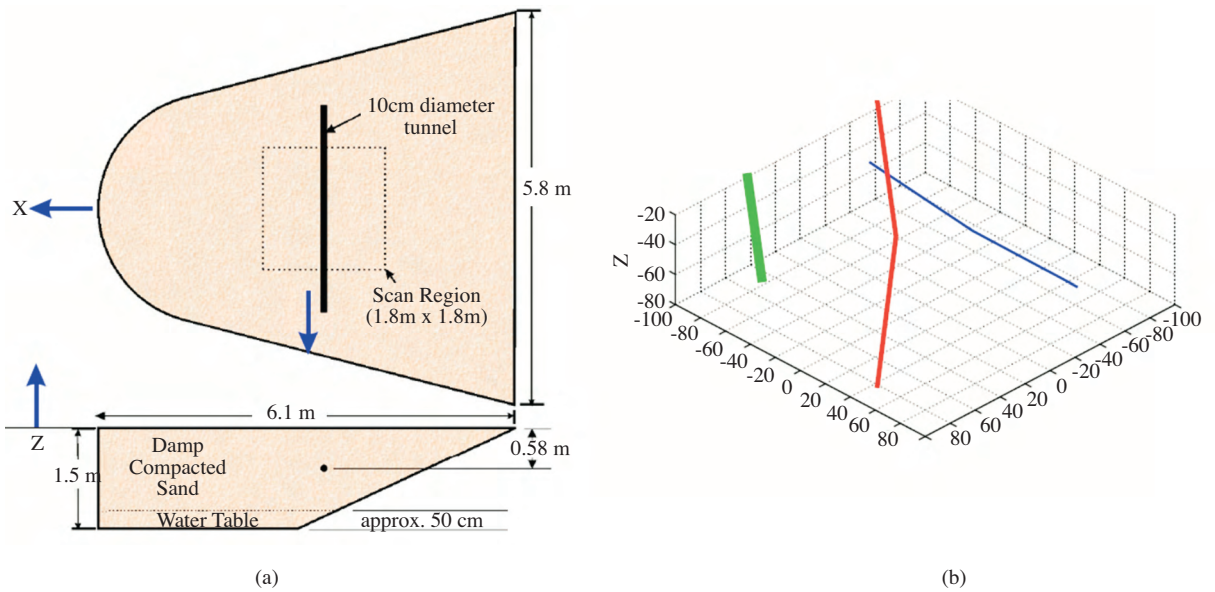

Figure 5. (a) Model of the sand tank with one 4" tunnel buried approximately 58 to 60 cm below the surface. (b) Layout of the three buried PCV pipes. The targets are a 2" pipe buried approximately 30 cm deep, a 1" pipe buried diagonally from the surface down to 60 cm, and a 0.5" pipe buried approximately 60 cm deep. The coordinate axes are the same in (a) and (b).

For both scenarios Figure 6 shows isosurface images for the backprojected subsurface GPR data [35]. Both the HT and the ARST algorithms are applied to the 3D subsurface images. Resolutions of 2° for (θ, φ) and 0.02 m for u and v are used for both algorithms. The ARST method used $\sigma = 0.15$ m. The detected lines are shown in Figure 6. The estimated parameters and the total run times of the algorithms are listed in Table 3, which shows that the ARST method can find the buried features much faster but just as accurately as the HT.

Table 3. Experimental results with 3D data comparing the ARST to the HT

Sensors	Scenario 1		Scenario 2	
Algorithms	ARST	HT	ARST	HT
θ (°)	96.68	96	(40.1, 44.5, 86.7)	(40, 45, 85)
φ (°)	0.08	0	(−4.4, 8.7, −1.3)	(−3,9,0)
u	0.162	0.15	(−85.2, −16.71, 55.99)	(−80, −16, 56)
v	0.620	0.62	(35.28, 35.79, 65.22)	(32, 36, 64)
Time (s)	1.44×10^3	5.74×10^4	1.78×10^3	5.79×10^4

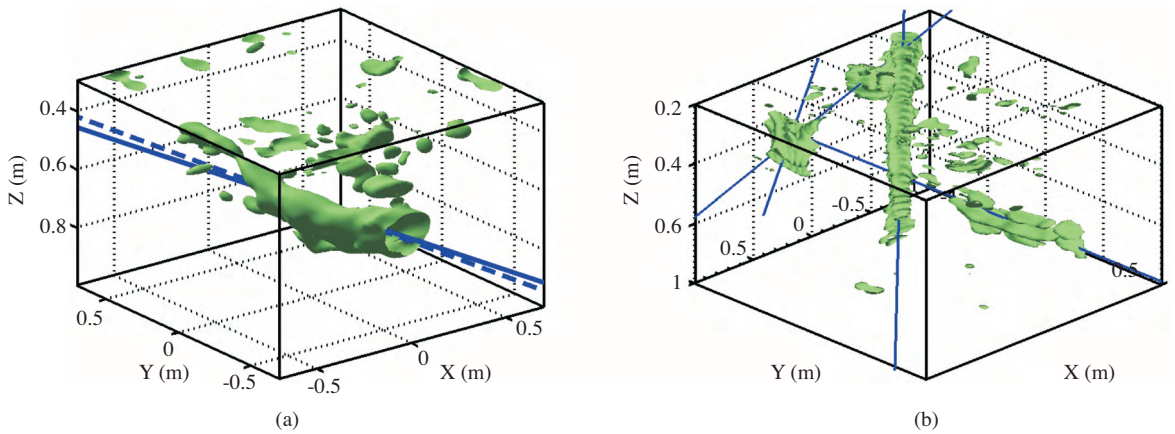

Figure 6. (a) 3D iso-figures at −15 dB for the 4" tunnel using the GPR sensor measurements. Dashed and solid lines show the detected lines for the HT and ARST methods, respectively.

4. Conclusions

An adaptive random sampling algorithm is introduced for detecting parameterized features. The random parameter samples are drawn from an updated distribution resulting in fewer required samples to detect a feature. Results from simulated and experimental data sets show that the proposed method has similar performance as the HT, and much less computing time. The proposed algorithm also outperforms faster HT variant algorithms such as RHT and PHT. The proposed algorithm is well suited for applications such as detecting pipes, tunnels or other features in subsurface images from seismic or GPR sensors.

References

[1] R. Gonzales, R. Woods, Digital Image Processing, Prentice-Hall, 2002.

[2] J. Illingworth, J. Kittler, A survey of the Hough Transform, Computer Vision, Graphics and Image Processing 44 (1988) 87–116.

[3] S. R. Deans, The Radon Transform and some of its applications, Krieger Publishing, 1993.

[4] H. Bakircioglu, E. Gelenbe, T. Kocak, Image Enhancement and Fusion with the Random Neural Network, Turkish Journal of Electrical Engineering and Computer Science, 5 (1997) 65–77.

[5] L. G. Shapiro, G. C. Stockman, Computer Vision, Prentice-Hall, 2001.

[6] A. C. Copeland, G. Ravichhandran, M. M. Trivedi, Localized Radon transform-based detection of linear features in noisy images, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994, pp. 664–667.

[7] M. Ekinci, E. Gedikli, Silhouette Based Human Motion Detection and Analysis for Real-Time Automated Video Surveillance, Turkish Journal of Electrical Engineering and Computer Science, 13 (2005) 199–229.

[8] A. Dell'Acqua, A. Sarti, S. Tubaro, L. Zanzi, Detection of linear objects in GPR data, Elsevier Signal Processing 88 (2004) 785–799.

[9] P. Toft, K. V. Hansen, Fast Radon transform for detection of seismic reflections, in: EURASIP EUSIPCO94, Vol. 1, 1994, pp. 229–232.

[10] P. V. C. Hough, A Method and Means for Recognizing Complex Patterns, in: US Patent 3069654, 1962.

[11] G. Beylkin, Discrete Radon Transform, IEEE Trans. Acoustic, Speech, and Signal Processing 35 (1987) 162–172.

[12] R. O. Duda, P. E. Hart, Use of Hough transformation to detect lines and curves in pictures, in: Comm. ACM, Vol. 15, 1972, pp. 11–15.

[13] D. H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition 13 (1981) 111–122.

[14] P. Toft, Using the generalized Radon transform for detection of curves in noisy images, in: ICASSP-2006, Vol. 4, 1996, pp. 2219–2222.

[15] P.Toft, The Radon transform theory and implementation, Ph.D. thesis, Technical University of Denmark, Lyngby, Denmark (1996).

[16] N. Kiryati, Y. Eldar, A. M. Bruckstein, A probabilistic Hough transform, Pattern Recognition 24 (1991) 303–316.

[17] L. Xu, E. Oja, P. Kultanan, A new curve detection method: Randomized Hough Transform (RHT), Pattern Recognition Letters 11 (1990) 331–338.

[18] L. Xu, E. Oja, Randomized Hough Transform (RHT): Basic mechanisms, algorithms, and computational complexities, CVGIP: Image Understanding 57 (1993) 131–154.

[19] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Comm. ACM 24 (1981) 381–395.

[20] R. Liu, Z. Ruan, S. Wei, Line detection algorithm based on random sample theory, in: SPIE Second International Conf. on Image and Graphics, Vol. 4875, 2002.

[21] T. Chen, K. Chung, A new randomized algorithm for detecting lines, Real Time Imaging 7 (2001) 473–481.

[22] J. Matas, C. Galambos, J. Kittler, Robust detection of lines using the progressive probabilistic Hough transform, Computer Vision and Image Understanding 78 (2000) 119–137.

[23] N. Kiryati, H. Kalviainen, S. Alaoutinen, Randomized and probabilistic Hough transform: unified performance evaluation, Pattern Recognition Letters 21 (2000) 1157–1164.

[24] H. Li, M. A. Lavin, R. J. L. Master, Fast Hough transform: A hierarchical approach, Computer Vision, Graphics and Image Processing 36 (1986) 139–161.

[25] J. Illingworth, J. Kittler, The adaptive Hough transform, IEEE Trans. Pattern Analysis and Machine Intelligence 9 (5) (1987) 690–698.

[26] C. Olson, Locating geometric primitives by pruning the parameter space, Pattern Recognition 34 (2001) 1247–1256.

[27] T. Breuel, Finding lines under bounded error, Pattern Recognition 29 (1) (1996) 167–178.

[28] D. Chai, Q. Peng, Image feature detection as robust model fitting, in: ACCV 7th Asian Conference on Computer Vision, Vol. 3852, 2006, pp. 673–682.

[29] W. R. Scott, Jr., T. Counts, G. D. Larson, A. C. Gurbuz, J. H. McClellan, Combined ground penetrating radar and seismic system for detecting tunnels, in: IEEE IGARSS, 2006, pp. 1232–1235.

[30] P. Gamba, S. Lossani, Neural detection of pipe signatures in ground penetrating radar images, IEEE Trans. Geoscience and Remote Sensing 38 (2000) 790–797.

[31] A. C. Gurbuz, J. H. McClellan, W. R. Scott, Jr., Compressive Sensing for Subsurface Imaging using Ground Penetrating Radars, Signal Processing, 89 10 (2009) 1959-1972.

[32] Y. Chen, Q. Yang, Y. Gu, J. Yang, Detection of roads in SAR images using particle filter, in: Proc. of ICIP, Atlanta, GA, 2006, pp. 2337–2340.

[33] W. R. Scott, Jr., K. Kim, G. D. Larson, A. C. Gurbuz, J. H. McClellan, Combined seismic, radar, and induction sensor for landmine detection, in: Geoscience and Remote Sensing Symposium, Vol. 3, 2004, pp. 1613–1616.

[34] A. C. Gurbuz, J. H. McClellan, W. R. Scott, Jr., A Compressive Sensing Data Acquisition and Imaging Method for Stepped Frequency GPRs, IEEE Trans. Signal Processing, 57 7 (2009) 2640–2650.

[35] T. Counts, G. D. Larson, A. C. Gurbuz, J. H. McClellan, W. R. Scott, Jr., Investigation of the detection of shallow tunnels using electromagnetic and seismic waves, in: Proc. SPIE, Vol. 6553, May 2007.