PLANE SWEEP METHOD FOR OPTIMAL LINE FITTING IN TRACK-BEFORE-DETECT

Yanmei Guo and Langford White

School of Electrical and Electronic Engineering, The University of Adelaide, South Australia

ABSTRACT

The Hough transform line detection method has been established as a viable technique for track-before-detect (TBD). However, its basic operation of binning and accumulating votes in the parameter space is computationally expensive. A more critical weakness of Hough transform is its dependence on parameter tuning (e.g., bin size and various thresholds), which can be non-intuitive and data-dependent. This leads to low detection rates in data with low signal-to-noise ratio and significant clutter. In this paper we propose a line detection algorithm with *guaranteed* global optimality for TBD. Our algorithm is based on the plane sweep algorithm for robust linear regression, with novel modifications to ensure its applicability under the TBD setting. Unlike the Hough transform, our algorithm has only one parameter to set (essentially the sensor false alarm rate) and can deterministically find the *best* solution according to a well-defined criterion. Simulation results on multi-dimensional TBD problems validate the accuracy and efficiency of our method.

Index Terms— track-before-detect, plane sweep.

1. INTRODUCTION

TBD refers to a tracking paradigm whereby measurements are accrued over an extended period such that target declaration is delayed until sufficient information is available. It is usually applied in low signal-to-noise ratio (SNR) scenarios, where a low first-threshold is applied to allow the accumulation of weak target signals (first-threshold crossings). Although this also yields significant false alarms, accumulating evidence over time enable the detection of faint targets.

A popular class of TBD algorithms are those based on the Hough transform (HT) [1, 2, 3, 4]. HT is widely used in image processing to detect (straight) line segments [5]. In tracking problems where the observation time is long and targets have low acceleration, the scan outputs tend to lie on a straight line when plotted against time [6]. There are also targets (e.g., large vessels in open water) that typically navigate using linear path segments. Such conditions can be exploited by HT to perform noncoherent integration over multiple scans [1].

Standard HT operates on 2D data. In the context of TBD, this usually implies that a single sensor measurement (e.g., range) is coupled with the time axis. The method builds a histogram in the 2D parameter space of lines (typically in polar representation), and each data point gives one vote for the bins corresponding to lines that pass through the point. The line corresponding to the bin with the highest vote count is regarded to have the strongest evidence to exist in the data.

To conduct multi-dimensional TBD, Moyer et al. [7] proposed to take two dimensions of the multi-dimensional data at a time, and apply standard HT to detect lines in each 2D subproblem. A gating process aggregates the multiple 2D line detections to yield an overall result. Fig. 1(a) depicts the idea.

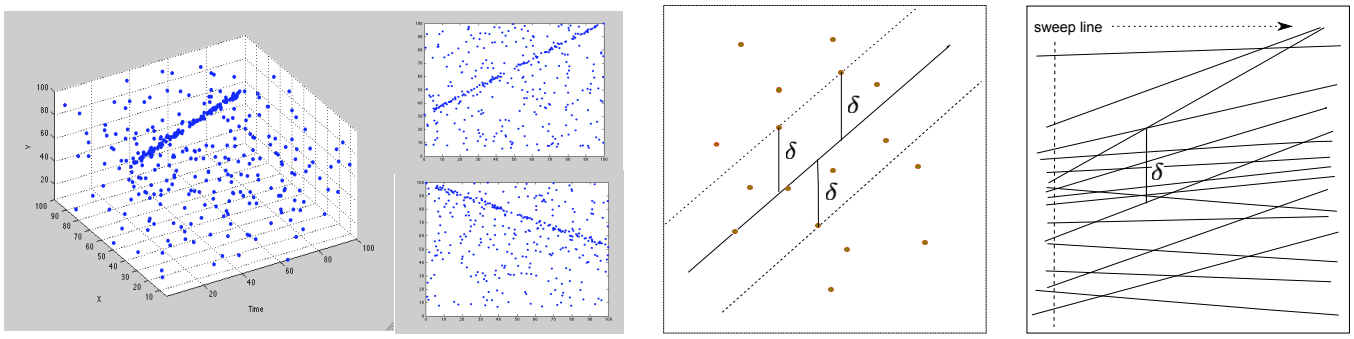
A well-known practical limitation of HT is the high sensitivity of the method to parameter settings. Most notably, the ideal binning resolution of the histogram is very dependent on the actual data characteristics. Various thresholds must also be tuned, e.g., the minimum amount of votes before a detection is declared. Finally, heuristic postprocessing must be conducted to achieve the intended result, e.g., to suppress redundant detections from neighbouring bins with equal votes. These issues make HT impractical in real-life systems.

In this paper, we propose a TBD line fitting algorithm with optimality guarantees. Our method is inspired by robust linear regression [8] based on the plane sweep method [9], but with novel modifications to ensure applicability under the TBD setting. Unlike HT, our algorithm deterministically and efficiently finds the *globally optimal* line (w.r.t. a well defined criterion) in a given input data. In addition, the plane sweep algorithm requires only a single parameter related to the sensor false alarm rate, which can be precalibrated in practical systems. When applied to Moyer et al.'s multi-dimensional TBD framework, our method exhibits much higher accuracy and dependability than previous techniques.

2. RELATION TO PRIOR WORK

Let there be N input points and $\rho \times \theta$ bins in the HT accumulator. In the binning phase, the HT algorithm iterates over all the points, where each point is converted via Radon transform to a sinusoid used to vote for the bins. In the detection phase, all the bins are examined to find the global maximum. Overall, the computational effort takes $\mathcal{O}(N\theta + \rho\theta)$ time.

Several authors have attempted to improve the usability of HT in TBD. Fan et al. [10] applied randomised Hough transform (RHT) [11] to reduce the computational burden of HT. Instead of creating a histogram, the parameter space is sam-



(a) (b) (c)

Fig. 1. (a) Illustrating TBD as a line fitting problem. The target has 0.8 probability of detection (PD), while the false alarm (FA) rate is 7×10^{-4}. The 3D data is projected down to 2D by pairing two dimensions at a time; two examples are shown here. (b) The least k-order robust regression for line fitting amounts to finding 3 equioscillating points with the smallest δ, where k points (here, $k = 9$) have residual $\leq \delta$. (c) The equivalent problem in the dual space can be solved efficiently by plane sweep.

pled by repeatedly fitting a line onto two randomly sampled data points. A linear array of bins is then maintained to accumulate the votes for lines that are sampled. While sampling alleviates high memory consumption, the introduction of randomness reduces the precision and repeatability of the results. A similar weakness affects Random Sample Consensus (RANSAC) [12], which uses a random sampling procedure to find the line with maximum consensus.

Beyond TBD, HT has been applied to a wide range of signal processing problems [14, 15, 16, 17, 18, 19]. However, HT is only one among a large body of methods for line detection. Our paper thus also serves to introduce a better line detection method to the signal processing community.

3. TBD WITH PLANE SWEEP

Moyer et al.'s [7] multi-dimensional reduction method allows us to focus on processing 2D data $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, where each $\mathbf{x}_i = [t_i, m_i]^T$ comprises of a sensor measurement m_i (e.g., range, bearing), and the time t_i at which the measurement was recorded (here, we always take t_i as time to remind the reader that the points may not lie in general position. In practice, t_i can be another measurement, and this does not invalidate our method below). Our goal is to fit a line onto $\mathcal{X}$, under the assumption that the target follows largely a linear path [1]. Note that there are also significant false alarms (outliers) in $\mathcal{X}$. Fig. 1(a) illustrates such a data.

3.1. Problem statement

A false alarm rate of ρ implies that $k = (1-\rho)N$ of the measurements in $\mathcal{X}$ arose from the target of interest. We aim to detect the line that passes through the k true detections. To measure the distance of points to lines, we designate

$$r_i(a,b) = |at_i + b - m_i| \tag{1}$$

as the *residual* of the i-th point $\mathbf{x}_i$ to the line defined by the parameters a and b. Our aim is to seek the a and b that solve

$$\min_{a,b} r_{(k)}(a,b), \tag{2}$$

where $r_{(k)}(a,b)$ is the k-th largest residual given a and b, i.e.,

$$r_{(j)}(a,b) < r_{(k)}(a,b) \qquad \forall\, j < k. \tag{3}$$

Problem (2) is an instance of the least k-th order estimator for robust regression [20]. We emphasise that the only required parameter k depends on the false alarm rate of the sensor.

It has been proven [8] that the solution is a line that *equioscillates* between three points — the three points have equal residual δ to the line, with one of the points on the other side of the line — and there are k points with residual less than or equal to δ to the line; see Fig. 1(b). Among all equioscillating lines, the one that minimises δ is the optimal solution to (2). One can envision a simple $\mathcal{O}(N^3)$ method that examines all triplets of points from $\mathcal{X}$. However, a more efficient scheme can be constructed by utilising duality.

3.2. Point and line duality

In HT the Radon transform is applied to map points to sinusoids. Here, a different kind of duality is employed. Define $D : (t,m) \mapsto (x,y)$ as a transformation from the measurement space (t,m) to the dual space (x,y). Specifically, D produces the point-to-line mapping

$$\mathbf{x} = [t,m]^T \mapsto D(\mathbf{x}) : y = tx + m. \tag{4}$$

Also, D conducts the following line-to-point mapping

$$\mathbf{l} : m = at + b \mapsto D(\mathbf{l}) = [-a,b]^T. \tag{5}$$

Observe that our aim to find the line parameters a and b that solves (2) is now reduced to finding a point in the dual space.

Under D, a point $\mathbf{x}$ that is determined by the two lines $\mathbf{l}_p$ and $\mathbf{l}_q$ is mapped to the line $D(\mathbf{x})$ that passes through the two points $D(\mathbf{l}_p)$ and $D(\mathbf{l}_q)$. Similarly, a line $\mathbf{l}$ that passes through the two points $\mathbf{x}_p$ and $\mathbf{x}_q$ is mapped to the point $D(\mathbf{l})$ that is determined by the two lines $D(\mathbf{x}_p)$ and $D(\mathbf{x}_q)$. Further, D preserves the vertical ordering between points and lines. More formally, if point $\mathbf{x}_p = [t_p, m_p]$ and line $\mathbf{l}_q : m = a_q t + b_q$ has (signed) vertical distance

$$\delta' = a_q t_p + b_q - m_p, \tag{6}$$

then it can be shown that the line $D(\mathbf{x}_p) : y = t_p x + m_p$ has (signed) vertical distance to point $D(\mathbf{l}_q) = [-a_q, b_q]^T$ of

$$\delta'' = b_q - (-t_p a_q + m_p) = \delta'. \tag{7}$$

More intuitively, if $\mathbf{x}_p$ is above $\mathbf{l}_q$ with distance δ', then $D(\mathbf{x}_p)$ is above $D(\mathbf{l}_q)$ by the same distance. These properties are crucial for transforming (2) to a more amenable problem.

Given $\mathcal{X}$, we first map each $\mathbf{x}_i$ to the line $\mathbf{l}_i : y = t_i x + m_i$ in the dual space. There, solving (2) is equivalent to finding vertical line segments that begin from the intersection of two lines and touches a third line, and where there are k lines that intersect with the line segments; Fig. 1(c) illustrates. The shortest of such vertical line segments is the solution to (2). This problem can be solved efficiently by plane sweep [8].

3.3. Plane sweep algorithm

The idea is to scan the dual space with a *sweep line (SL)*; see Fig. 1(c). When an intersection is found, we test whether there is a vertical line segment from the intersection to another line with k lines passing through the segment. Clever usage of data structures allows this to be done in $\mathcal{O}(N^2 \log N)$ time.

The original algorithm of [8], however, is not suitable for TBD, since it assumes that $\mathcal{X}$ is in general position, e.g., no two points share the same t value since this yields parallel lines in the dual space. In typical tracking systems, multiple measurements are produced in the same time instance, yielding multiple points with the same t values [10]. Subjecting $\mathcal{X}$ to the original algorithm of [8] would result in failure.

Here, we present our novel plane sweep algorithm tailored to the TBD problem; Algorithm 1 summarises our method. Similar to [8], our method employs two data structures LIST and HEAP. The former is a linked list that stores the $\mathbf{l}_i$'s based on their top-down order of crossing with SL. To initialise LIST, the $\mathbf{l}_i$'s are sorted increasingly in slope t_i. For lines which have the same t_i values (i.e., parallel lines), they are additionally sorted decreasingly in intercept m_i. After initialisation, the line with the smallest slope and largest intercept is at the front of LIST, while the line with the largest slope and smallest intercept is at the rear of LIST. This corresponds to placing SL at the left of all the intersections of the $\mathbf{l}_i$'s.

HEAP is a min-heap that stores a set of intersection points of the $\mathbf{l}_i$'s. It allows to retrieve in $\mathcal{O}(1)$ time the leftmost intersection point contained therein, and to insert or delete intersection points in $\mathcal{O}(\log N)$ time. To initialise HEAP, we traverse the initial LIST and calculate for each neighbouring $\mathbf{l}_i$'s their intersection and insert it into HEAP. Parallel lines are simply skipped. This ensures that the leftmost intersection point exists in HEAP before the main operations begins.

In the rest of the algorithm, we repeatedly remove the leftmost intersection point in HEAP - this simulates the visiting of SL on the intersection point. Given a removed point, we identify the two neighbouring lines $\mathbf{l}_p$ and $\mathbf{l}_q$ in LIST that gave rise to the point. We then find (if exist) the k-th line before and after the position of $\mathbf{l}_p$ and $\mathbf{l}_q$ in LIST, and calculate their vertical distance to $\mathbf{l}_p$ and $\mathbf{l}_q$. This corresponds to finding a vertical segment with the desired property indicated in Fig. 1(c). The shortest segment found thus far is recorded.

After the above operation, the SL is considered to have moved past the point. This corresponds to a swap in the position of $\mathbf{l}_p$ and $\mathbf{l}_q$ in LIST. New intersection points are then generated by the pairing of $\mathbf{l}_p$ and $\mathbf{l}_q$ with their respective new neighbours (if exist, and ignoring lines that are parallel). The new points are then inserted into HEAP and the process continues. The procedure ensures that the next point to be removed from HEAP is always the closest one to the right of the SL. The algorithm terminates when HEAP is empty.

Algorithm 1 Plane sweep algorithm for TBD.

Require: Data $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ where $\mathbf{x}_i = [t_i, m_i]^T$, sensor false alarm rate ρ.

```
k ← (1 − ρ)N.
For all i, set l_i as line with slope t_i and intercept m_i.
Store l_i's in linked-list LIST, then sort LIST increasingly by t_i, then decreasingly by m_i for tie-breaking.
HEAP ← empty min-heap.
δ ← ∞.
for p = 1, 2, ..., N − 1 do
    if p-th and (p + 1)-th items in LIST not parallel then
        Calculate intersection point and insert into HEAP.
    end if
end for
while HEAP not empty do
    Remove leftmost point d from HEAP.
    for each l = {l_p, l_q} which gave rise to d do
        δ' ← distance of line k positions before l in LIST.
        δ ← min(δ, δ').
        δ' ← distance of line k positions after l in LIST.
        δ ← min(δ, δ').
    end for
    Swap l_p and l_q in LIST.
    For newly adjacent lines in LIST which are not parallel, calculate intersection and insert into HEAP.
end while
return Centre position of vert. segment with smallest δ.
```

4. SIMULATIONS

Our simulation data is generated as follows: for each data instance, a line equation in 3D is randomly generated. One of the axis is chosen as the time dimension with T number of scans, while the other two dimensions define a 100×100-unit range. A point is generated on the line at each scan with probability equal to PD. False measurements are then produced randomly in each scan according to the FA probability. We corrupt the points with Normal noise of standard deviation σ. This produces an instance of a 3D data $\{t_i, x_i, y_i\}_{i=1}^N$. See Fig. 2 (left) for an example with $T = 100$.

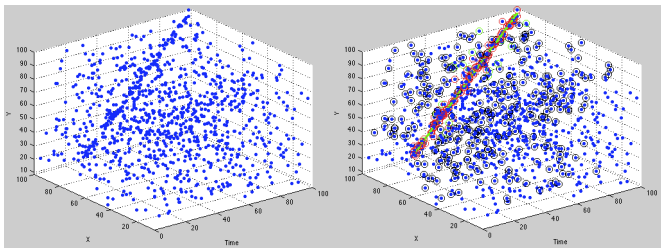


Fig. 2. (left) Input data; (right) Final gating outcome. Among the line detection results in the 3 combinations of pairwise dimensions, points circled in black were chosen once as inliers, green were chosen twice, and red were chosen thrice.

Following [7], we take two dimensions the 3D data at a time, i.e., $\{t_i, x_i\}_{i=1}^N$, $\{t_i, y_i\}_{i=1}^N$ and $\{x_i, y_i\}_{i=1}^N$, and conduct line fitting in each combination. We run and compare four algorithms: HT, RHT, RANSAC and Plane Sweep (PS). More specifically, each algorithm is executed to find the line at each 2D combination, and the inliers lying on the detected line are identified. The gating procedure of [7] is then employed to aggregated the inliers selected to obtain the final result; see Fig. 2 (right) for an example.

To objectively compare the accuracy of the methods, we introduce a scoring scheme for the final gating result. For a respective method, after gating, each point $\{t_i, x_i, y_i\}$ has a count $c_i \in \{0, 1, 2, 3\}$ corresponding to the number of times it was selected by the line detection processes as an inlier. Each point also has the ground truth label

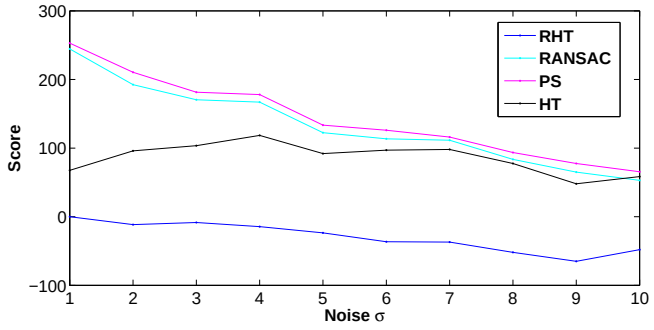
$$l_i = \begin{cases} 1, & \{t_i, x_i, y_i\} \text{ is an inlier,} \\ -1, & \text{otherwise.} \end{cases} \tag{8}$$

The score of a particular method is then calculated as

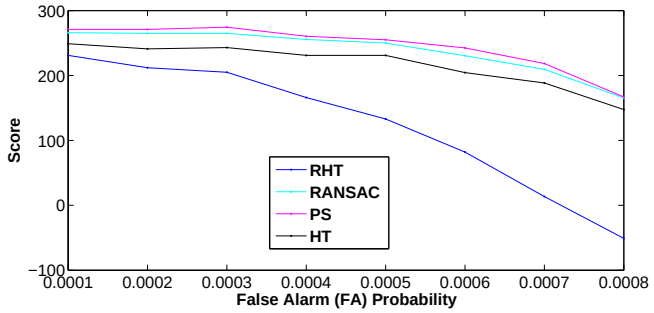
$$\sum_i c_i l_i, \tag{9}$$

where higher scores indicate higher line fitting accuracies.

To test the effectiveness of the algorithms under varying difficulties, we control the σ values and FA probabilities. In the first experiment, we fixed $T = 100$, PD $= 0.8$, FA probability $= 7 \times 10^{-4}$, and varied $\sigma = 1, 2, \ldots, 10$. For each



(a)



(b)

Fig. 3. Comparison of final gating results of 4 methods under varying σ and FA probability. Higher scores are better.

σ value, 1000 data instances were generated and the aforementioned procedures were executed. Fig. 3(a) shows the average scores across the data instances of each method plotted against σ. We repeated the above experiment by fixing $T = 100$, PD $= 0.8$, $\sigma = 6$, and varying FA probability $= \{1, 2, \ldots, 8\} \times 10^{-4}$. Fig. 3(b) shows the result.

The results clearly show that PS outperforms the other three algorithms (HT, RHT and RANSAC), owing to its guarantee to return the globally optimal parameters in line detection. Although RANSAC is not far behind, its inherent randomness is undesirable for applications that require high repeatability and dependability. Note that the score (9) is fair since none of the algorithms directly optimise this score. In terms of run time, although PS is slower than the other randomised methods, in all the data instances Algorithm 1 terminated within 2 seconds. Note that the average number of points N of the data instances is 950.

5. CONCLUSIONS

In this paper we have proposed an alternative line-detection method for TBD based on plane sweep algorithm. We showed that our method is more accurate compared to the more established methods (HT, RHT and RANSAC). This has opened up more interesting questions on real time and field usage of PS algorithm on TBD. We will investigate more on continuous multi-target application of PS algorithm in the future.

6. REFERENCES

[1] B. D. Carlson, E. D. Evans, and S. L. Wilson, "Search radar detection and track with the Hough transform, part I: system concept," *IEEE Trans. Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 102–108, 1994.

[2] B. D. Carlson, E. D. Evans, and S. L. Wilson, "Search radar detection and track with the Hough transform, part II: detection statistics," *IEEE Trans. Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 109–115, 1994.

[3] B. D. Carlson, E. D. Evans, and S. L. Wilson, "Search radar detection and track with the Hough transform, part III: detection performance with binary integration," *IEEE Trans. Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 116–125, 1994.

[4] G. A. Richards, "Application of the Hough transform as a track-before-detect method," in *IEE Colloq. on Target Tracking and Data Fusion*, 1996.

[5] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Computer Vision, Graphics, and Image Processing*, vol. 44, pp. 87–116, 1988.

[6] L. Wang, R. Tao, and S. Zhou, "Detection and imaging of slowly moving target of airborne SAR based on the GMCWD-Hough transform," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2003.

[7] L. R. Moyer, J. Spak, and P. Lamanna, "A multi-dimensional Hough transform-based track-before-detect technique for detecting weak targets in strong clutter backgrounds," *IEEE Trans. Aerospace and Electronic Systems*, vol. 47, no. 4, pp. 3062–3068, 2011.

[8] D. L. Souvaine and J. M. Steele, "Time- and space-efficient algorithms for least median of squares regression," *Journal of the American Statistical Association*, vol. 82, no. 794–801, 1987.

[9] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational geometry: algorithms and applications*, Springer, 2008.

[10] L. Fan, X. Zhang, and L. Wei, "TBD algorithm based on improved randomized Hough transform for dim target detection," *Progress in Electromagnetics Research C*, vol. 31, pp. 271–285, 2012.

[11] P. Kultanen, L. Xu, and E. Oja, "Randomized Hough transform (RHT)," in *Int. Conf. on Pattern Recognition*, 1990.

[12] V. Cevher, F. Shah, R. Velmurugan, and J. H. McClellan, "A multi target bearing tracking system using random sampling consensus," in *IEEE Aerospace Conference*, 2007.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communicatinos of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[14] J. Perkins and I. Coat, "Pulse train deinterleaving via the Hough transform," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 1994.

[15] R. Suleesathira and L. F. Chaparro, "Interference mitigation in spread spectrum using discrete evolutionary and Hough transforms," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2000.

[16] D. L. Robie and R. M. Mersereau, "The use of Hough transforms in spatial error concealment," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2000.

[17] S. Thayilchira and S. Krishnan, "Detection of linear chirp and non-linear chirp interferences in a spread spectrum signal by using Hough-Radon transform," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2002.

[18] L. A. Cirillo, A. M. Zoubir, and M. G. Amin, "Direction finding of nonstationary signals using a time-frequency Hough transform," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2005.

[19] L. A. Cirillo, A. M. Zoubir, and M. G. Amin, "Estimation of FM parameters using a time-frequency Hough transform," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2006.

[20] P. J. Huber, *Robust statistics*, John Wiley & Sons Inc., 2nd edition, 2009.