Dungan, K. E. and Potter, L. C., "Classifying sets of attributed scattering centers using a hash coded database," in Algorithms for Synthetic Aperture Radar Imagery XVII, Zelnio, E. G. and Garber, F. D., eds., Proc. SPIE 7699 (2010).

Classifying Sets of Attributed Scattering Centers using a Hash Coded Database

Kerry E. Dungan, Lee C. Potter

Dept. of Electrical & Computer Engineering, The Ohio State University, Columbus, OH;

ABSTRACT

We present a fast, scalable method to simultaneously register and classify vehicles in circular synthetic aperture radar imagery. The method is robust to clutter, occlusions, and partial matches. Images are represented as a set of attributed scattering centers that are mapped to local sets, which are invariant to rigid transformations. Similarity between local sets is measured using a method called pyramid match hashing, which applies a pyramid match kernel to compare sets and a Hamming distance to compare hash codes generated from those sets. By preprocessing a database into a Hamming space, we are able to quickly find the nearest neighbor of a query among a large number of records. To demonstrate the algorithm, we simulated X-band scattering from ten civilian vehicles placed throughout a large scene, varying elevation angles in the 35 to 59 degree range. We achieved better than 98 percent classification performance. We also classified seven vehicles in a 2006 public release data collection with 100% success.

Keywords: SAR, scattering centers, pyramid match hashing, point pattern matching

1. INTRODUCTION

Identification of civilian vehicles in all-weather persistent surveillance of a large scene is enabled by wide aperture synthetic aperture radar (SAR) imagery. In this paper, we present a new algorithm, based on attributed scattering centers,[1] that finds the closest entry of a training vehicle database for a vehicle query. To accomplish this, we combine three concepts: mapping attributed scattering centers to sets that are invariant to rotations and translations,[2,3] applying a method called pyramid match hashing[4] (PMH), and performing a simple fusion[5,6] of results for individual scatterers.

Figure 1 displays an outline for our algorithm. In an offline step, wide aperture circular SAR training imagery is abstracted to attributed scattering centers (or point patterns). These points have 2D location information, and hence vary with respect to rotations and translations in a plane. The invariant mapping block describes each scattering center by a local neighborhood of points in a local frame of reference described by the point and its azimuth to the radar; as a result, the scattering centers are represented as a set of sets. Subsequently, the PMH algorithm represents each local set with a binary hash code for fast digital processing.

As shown in Figure 1, a query image is subjected to the same extraction, invariance, and PMH steps during online processing. Each hash code from the query is compared to the hash codes of the database using a Hamming

Please send author correspondence to K.E.D. E-mail: dungan.2@osu.edu.

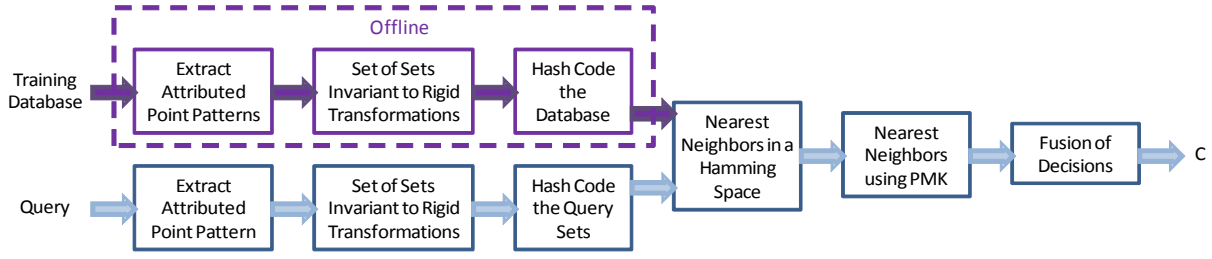


Figure 1. The classification algorithm as two input branches, where the training data is processed offline and the query is processed when data is available. Query hash codes are compared against the database, and the nearest several candidates from the database are evaluated using the pyramid match kernel for classification.

distance to find a set of nearest neighbors (NN), which are candidate nearest neighbors. This reduced set of candidates is further processed by the more sensitive pyramid match kernel[7] (PMK); the PMK is our measure of similarity in comparing the invariant, local, attributed point patterns. Finally, each point (local set) of the query is assigned a label from the training set, and we implement a majority vote or mode decision for classification.

We created two experiments to demonstrate the capabilities of our classification algorithm. The first data set contains synthetic variations of ten civilian vehicles that are placed in a large scene. This simulation was first used by Dungan and Potter[8] based on synthetic data domes in the CVDomes data set.[9] For a second experiment, a 2006 challenge problem[10] provided phase history for eight circular orbits of a parking lot containing seven civilian vehicles. We extracted imagery for the seven vehicles and classified vehicles from the last seven orbits, using the first orbit for training data. As with CVDomes data, the 2006 data set was previously exploited and used in classification experiments.[11]

The invariant PMH based method of Figure 1 is fast compared to the minimized least trimmed squares Hausdorff distance method[8,11,12] of our previous work. Single threaded query times for the 10-class synthetic data set reduced from 20 minutes to less than 10 seconds, while performance improved from 96.5% (97.7% in a submitted manuscript) to 98.7%. We also achieved a similar two orders of magnitude decrease in processing time for the 2006 data set; the performance numbers showed the same 100% classification.

The CVDomes based experiment was designed to be more difficult than the 2006 based experiment. By moving a vehicle to different locations in a large scene, variations in elevation to the radar from 35 to 59 degrees cause vehicle signatures to differ primarily due to layover[13] changes.

The remaining sections of the paper are organized as follows. First, in Section 2, we describe the extraction of attributed point patterns and the formation experiments using the CVDomes and 2006 data sets. Next, in Section 3, we describe the pose invariant mapping of point patterns along with Mahalanobis-based distances and whitening of the data. Then, in Section 4, we briefly describe the PMH algorithm, implementation, and decision process. Finally, in Section 5, we show classification results for the CVDomes and 2006 data sets.

2. CIVILIAN VEHICLE CIRCULAR SAR DATABASES

In this section, we describe the design of two databases that are used to evaluate the classification algorithm of Figure 1. The first data set is synthetically generated imagery using ten vehicles of the CVDomes data set.[9] The second database contains imagery for seven vehicles that were extracted from the parking lot of the 2006 challenge problem.[10]

2.1 Experiment Design using CVDomes

In circular SAR, the radar, along with its ground projection, orbit the origin in a circular path. The angle of elevation from a vehicle to the radar varies throughout the orbit, except for a vehicle at the origin. For our experiments, we assume the radar is orbiting with an elevation of 45 degrees with respect to the scene center. Furthermore, we assume the terrain is flat asphalt across the patch, resulting in elevation angles between 35 and 59 degrees.

To generate imagery from the phase history data in the CVDomes data set,[9] we used fast backprojection[14,15] over a 5 degree aperture and generated 120 images around the SAR orbit with an 40 percent overlap. Figure 2 shows example images for a Maxima (v5), where all 120 images are coherently added to give a 360 degree representation of a vehicle. When generating 2D SAR imagery of a vehicle on asphalt, it is typically possible to visualize an inner and outer ring of scattering. The inner ring is generated by radar echoes that experience a two-bounce path between the asphalt and the side of the vehicle; thus, the inner ring represents time-of-flight to the base outline of the vehicle.[16] In contrast, the outer ring arises from radar echoes reflected by elevated portions of the vehicle, such as the roof line. These elevated features project onto the image in a direction orthogonal to the radar line of sight – an effect known as layover.[13] The polarization feature, ν, encodes even versus odd bounce echoes.[17–19]

Depending upon the pose and location of the vehicle within the scene, scattering features may migrate or not persist, and the elevated features will exhibit different layover as a function of elevation angle. Thus the same

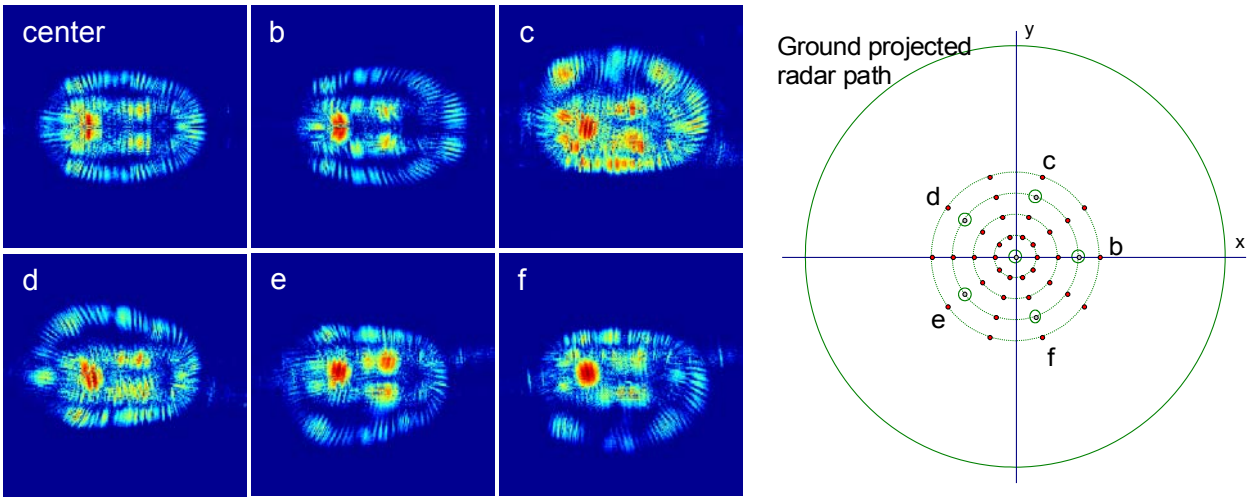


Figure 2. If a vehicle is offset from the center of a large circular SAR scene it will experience different elevations to the radar over the flight path. As a result, the same vehicle looks differnt when translated around the scene. We generated simulated vehicle imagery for each of the dots in the graph; imagery for six of the locations are displayed. The circled dots are locations used in the classification training set.

CVDomes Vehicles

1	2	3	4	5	6	7	8	9	10
Camry	Civic	Jeep93	Jeep99	Maxima	MPV	Mitsubishi	Sentra	Avalon	Tacoma

Table 1. The CVDomes data set contains simulations for ten civilian vehicles. In this paper, we refer to these vehicles with a number in one through ten.

vehicle in two different locations can appear significantly different, as shown in Figure 2. The top left image shows the vehicle centered in the scene, while the remaining images show the vehicle offset by 0.4 times the flight path radius in five different locations in the scene.

Based on the dots in the right half of Figure 2, we generated 50 sets of images for each of the ten vehicle classes in Table 1. There are ten 36-degree pose rotations at the center and ten locations on each of the four concentric rings about the center. Each vehicle location $v = ||v|| \, e^{j\theta_v}$, in polar coordinates, is represented by the pair (q, θ_v), where $q \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$ is the ratio $q = ||v|| \, / \, ||r_v||$ and $||r_v||$ is the flight path radius. The resulting database contains 500 sets of images. Each set of images contains 120, five degree, subaperture images with 40 percent overlap. The six circled locations in Figure 2 are set aside as training samples. Thus, samples in the set $\{(0.0, 0°), (0.3, 0°), (0.3, 72°), (0.3, 144°), (0.3, 216°), (0.3, 288°)\}$ are used for training, while the remaining 44 locations are used as classification samples.

The first step in applying the approach shown in Figure 1 is to process the SAR imagery into an attributed point pattern. The high-frequency scattering response of an object is well approximated as a sum of scattering centers,[20] and it is possible to discriminate amongst various objects using an unordered set of feature points.[8,11,21–25] These attributed scattering centers or feature points can include location, amplitude, polarization, and a variety of other attributes.[1] Although radar bandwidth limits the ability to distinguish point sets from two vehicles observed from a given viewing angle,[26] the fusion of a diversity of vehicle aspects can greatly enhance discrimination.[27–29]

For each image in each set, we implemented a simple peak extractor to approximate scattering center locations. Each scattering center is represented as a vector containing the x and y location, the center azimuth, θ, of the associated imaging aperture, the amplitude, A, of the peak, and an even/odd bounce bit ν to encode polarization.[8,17–19] Thus each point in an image is represented by

$$\vec{p} = (x, y, \theta, A, \nu). \tag{1}$$

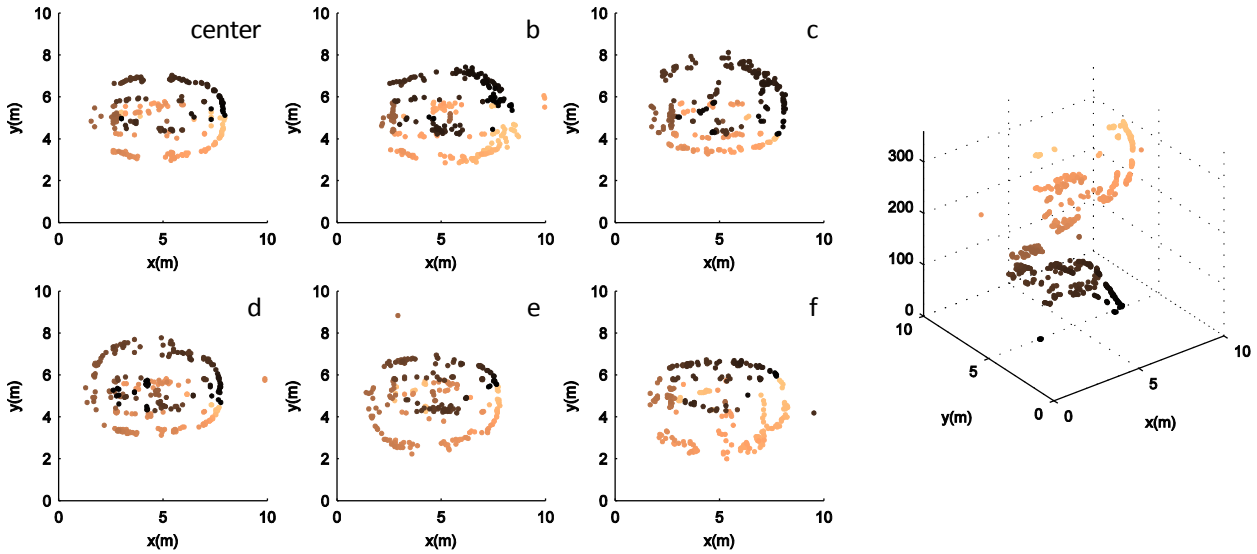


Figure 3. By extracting attributed scattering centers from the imagery shown in Figure 2, vehicles are abstracted to a set of attributed points. Among the attributes for a scattering center are 2D location and azimuth to the radar. Darker points in the plots start with a zero degree azimuth in the x direction; lighter points represent increasing azimuths as the radar orbits counterclockwise from the x-axis. The plot to the right shows a spiral pattern by representing the azimuth on a z-axis.

2006 Data						
1	2	3	4	5	6	7
Prizm	Maxima	Sentra	Santa Fe	Camry	Taurus	Malibu

Table 2. The 2006 data collection contains seven identifiable civilian vehicles. In this paper, we refer to these vehicles with a number in one through seven.

A vehicle sample is represented by a set

$$P = \{\vec{p}_1, \cdots, \vec{p}_n\} \tag{2}$$

containing all of the attributed scattering centers in the collection of 120 subaperture images. Figure 3 shows example views of $\{x, y, \theta\}$ for the images of Figure 2. As azimuth increases counterclockwise from the x-axis, the azimuth attribute goes from dark to light points. The 3D plot shows the azimuth on the z-axis, and the leading edge (to the radar) of the vehicle is apparent by the spiral pattern.

2.2 Experiment Design using 2006 Collection

A 2006 data collection[10] generated circular SAR phase history for a parking lot scene containing seven characterized civilian vehicles. The data set contained information for eight orbits of the radar. The top left image in Figure 4 shows an image of the parking lot with numeric labels for the seven vehicles, while Table 2 cross-references the label numbers with vehicle model.

Using the same imaging technique from Section 2.1, we isolated the seven vehicles as shown by the seven remaining images of Figure 4. Each image is a sum of 120, five degree aperture, backprojection images with 40 percent overlap around the SAR orbit. To generate attributed point patterns, we extracted the same information as described with Equation (1).

For the experiment, the attributed point patterns from the first orbit of the SAR are used as training data. Then, in the subsequent seven passes, the vehicles are identified based on the training data.

3. INVARIANT MAPPING OF ATTRIBUTED POINT PATTERNS

A set of attributed scattering centers describes a class sample; this set of attributed points is considered equivalent to another set under rigid transformations for purposes of pattern recognition. In other words, a translated

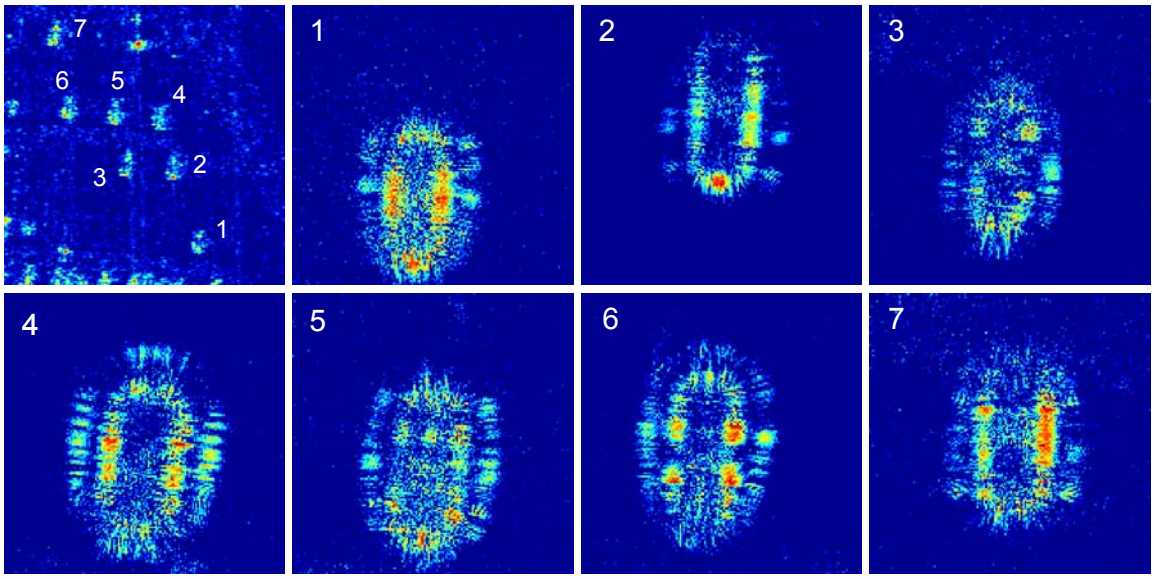


Figure 4. There are seven identifiable civilian passenger vehicles in 2006 data collection parking lot. The top left image shows the seven vehicles in one scene, while the remaining images show close-up views of each vehicle independently. We extracted attributed scattering centers of these seven vehicles for all eight orbits of the radar.

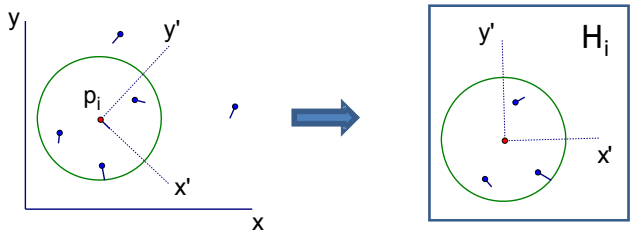


Figure 5. An example attributed point pattern with an anisotropy (azimuth) attribute is displayed on the left. A point p_i is represented by a set H_i containing the point along with three nearest neighbors in a local frame of reference.

or rotated vehicle is the same vehicle. However, matching sets of points under rigid transformations is computationally intensive. For example, Dungan and Potter applied a nonlinear optimization called a particle swarm to minimize a distance between point patterns.[8,11]

To avoid the nonlinear optimization, some researchers have aligned point patterns using an anisotropic attribute.[2,3] We have adapted a variation of local alignment to the SAR recognition problem. Let $P = \{p_1, p_2, \cdots, p_n\} \subset \mathbb{R}^f$ represent a class sample, where f is the number of attributes (i.e. $f = 5$ from Equation (1)) and n is the number of points in a class sample. Next, we proceed with invariant transformation and whitening steps.

First, as illustrated in Figure 5, we represent each point in P by a set of its K nearest neighbors. The distance between points is measured using a Mahalanobis distance,

$$d(p_i, p_j) = \sqrt{(p_i - p_j)^T \Sigma^{-1} (p_i - p_j)}, \tag{3}$$

where Σ is the measurement error covariance matrix. A local origin centers on the point, and the direction of the azimuth attribute defines a local x'-axis with the y'-axis 90 degrees counterclockwise. Each of the x, y, and θ attributes for the K nearest neighbors of p_i is transformed to the local frame of reference (A and ν attributes do not change). Thus, each $p_i \in P$ is represented as a set of $K + 1$ points.

Secondly, the transformed local sets are whitened to give an identity error covariance, a desirable property for the PMK of the next section. Sets are whitened using the transformation W,

$$\Sigma = V \Lambda V^T \Rightarrow W = V \Lambda^{-\frac{1}{2}} V^T. \tag{4}$$

Notice that the local transformation process aligns nearest neighbor points to a frame of reference that is noisy. This introduces additional error; thus, the whitening process is approximate.

In summary, each $p_i \in P$ is mapped to a transformed and whitened set along with its K nearest neighbors,

$$p_i \rightarrow H_i = \{h_{\xi_0^{(i)}}, h_{\xi_1^{(i)}}, \cdots, h_{\xi_K^{(i)}}\}, \tag{5}$$

where $\{\xi_0^{(i)}, \xi_1^{(i)}, \cdots, \xi_K^{(i)}\}$ is a sequence of indexes in P to the K nearest neighbors of the i^{th} element of P. Thus P is mapped to the set of sets H,

$$P = \{p_1, p_2, \cdots, p_n\} \rightarrow H = \{H_1, H_2, \cdots, H_n\}. \tag{6}$$

As a result, each point of each class sample is represented as a set that is considered a database entry. The training database $\mathcal{H}$ consists of a union of all H mappings for the class samples, creating one big database. Thus, $\mathcal{H}$ is a set of sets containing an entry for every point of every training sample; it is necessary to maintain an index cross referencing members $\mathcal{H}_i \in \mathcal{H}$ to a class label.

When accessing the database, a query is invariantly mapped to a set of sets $Q = \{Q_1, Q_2, Q_3, \cdots\}$, where the number of sets in Q is the same as the number of points in the query attributed point pattern.

4. PYRAMID MATCH HASHING AND FUSION OF RESULTS

The pyramid histogram,[7] pyramid match kernel,[7] and pyramid match hashing[4] are described in detail by Grauman and Darrell; here, we provide a brief overview of the concept along with insight into our implementation.

4.1 The Pyramid Histogram

To begin, the data must be represented as pyramid histograms, described as follows:

- For the base layer of a pyramid histogram, scale, shift, and quantize the $K+1$ points for a member of the database $\mathcal{H}$ in Equation 6 onto a discrete grid of integers in $\{1, \cdots, N_0\}$ for each of the f dimensions. The N_0^f discrete locations represent single counts in bins of a very sparse histogram;
- For subsequent layers, repeatedly decimate the grid by a factor of two in each dimension while accumulating the bin counts. Eventually all points in H are counted in a single bin;
- For $L = \lceil \log_2 N_0 \rceil$ layers, there are $(N_0/2^l)^f$ bins per layer for $l \in \{0, \cdots, L-1\}$.

4.2 The Pyramid Match Kernel

For a database entry $\mathcal{H}_i \in \mathcal{H}$ and a query $Q_j \in Q$, the PMK is a measure of similarity between two pyramid histograms given by

$$\mathcal{P}_\Delta(\Psi(\mathcal{H}_i), \Psi(Q_j)) = w_{L-1} I_{L-1} + \sum_{l=0}^{L-2} (w_l - w_{l+1}) I_l, \tag{7}$$

where Ψ is the pyramid histogram function, $w_l = 2^{-l}$ is a weighting function, and I_l is an intersection between the l^{th} layers of the two pyramid histograms. To create a Mercer kernel, the entries of weighting function must be monotonically decreasing, and w_l satisfies this condition.[7] The intersection between each level of two pyramid histograms $\Psi(\mathcal{H}_i)$ and $\Psi(Q_j)$, I_l, is defined as the sum of the minimum number in corresponding bins between the hypercubes at that pyramid level.

Our implementation quantizes the data, using $N_0 = 512$ and $f = 5$, to a 512^5 hypercube (histogram); the invariantly mapped databases for both the CVDomes and 2006 collection, consisting of each point of each class sample, adequately fits in this grid. The hypercube is large, and the complexity of calculating the PMK is based in large part on the time to intersect two hypercubes. However, the hypercubes are sparsely populated and the intersection is a matter of comparing populated indices; thus time complexity is on the order of the number of points in the sets $\mathcal{H}_i$ and Q_j as opposed to the size of the hypercube.

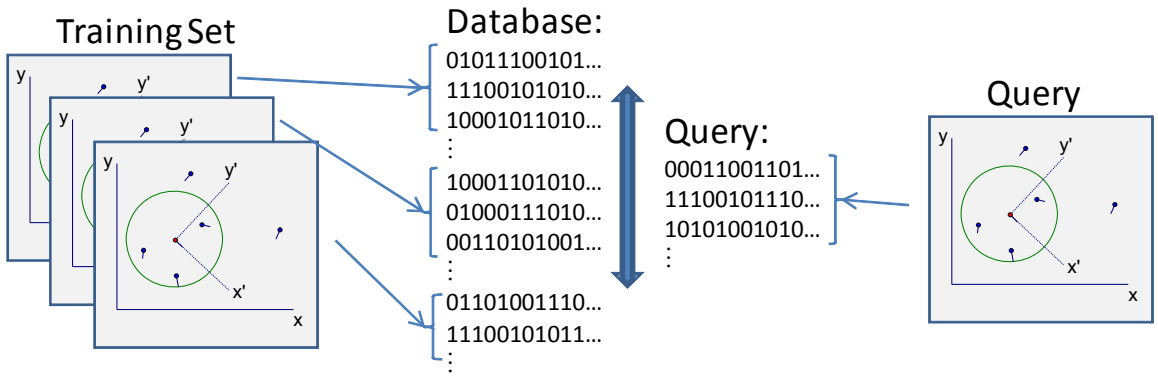


Figure 6. Each point of each training sample is mapped to a set of nearest neighbors in a local frame of reference. These sets are coded into M-bit hash keys. The points of a query sample are also coded to M-bit hash keys. The query codes are compared to database entries using Hamming distances.

4.3 Pyramid Match Hashing

The PMK can be used to implement a kernel-based classifier, such as a nearest neighbor, where the classifier chooses the largest kernel between Q_j and the database $\mathcal{H}$. However, the time complexity grows linearly in the number of elements in $\mathcal{H}$. The PMH algorithm, on the other hand, provides a faster method to preprocess the database into binary hash codes, where queries are compared to the database using Hamming distances (sum of XOR between binary codes). A nearest neighbor search still requires linear time complexity in calculating Hamming distances, but it is significantly faster than calculating the PMK. Furthermore, it is possible to perform an approximate nearest neighbor search in sublinear time,[4] if desired, for large databases.

An M-bit hash code y for a member of the database or a set in the query is generated from a 1-bit quantized map of the vectorized and square-root weighted pyramid histogram[4] x,

$$y = Q(Ax). \tag{8}$$

The 1-bit quantization, Q, is calculated using the sign of Ax, and the $A \in \mathbb{R}^{M \times N}$ matrix is a zero mean unit variance random matrix. The row count N is the size of the vectorized pyramid histogram, i.e.,

$$N = \sum_{l=0}^{L-1} \left(N_0 2^{-l}\right)^f. \tag{9}$$

Since A is very large, while x is sparse, we seed a fast Mersenne Twister[30] pseudorandom number generator with an index to the entry of A to generate values in A on demand.

4.4 Overview and Fusion of Decisions

Figure 6 shows an overview of the invariant mapping and hash coding process of Figure 1. The scattering centers of each class sample are represented by an f dimensional attributed point pattern. Each point of each class sample is mapped to a rotation and translation invariant set of its K nearest neighbors. These sets are then hash coded into a database. For example, 60 class samples with 200 points each would generate a database of $12,000$ hash codes.

Next, a query image is also hash coded using PMH to generate a set of hash codes for each point in the query pattern. As described in Figure 1, the query hash codes are compared to the database hash codes using Hamming distances. To preserve the measure of similarity/distance between sets in the PMK vs. Hamming distance, the hash code would need to be large. Thus, we limit the length of the hash code and calculate several nearest neighbors (or approximate nearest neighbors) using the Hamming distance. Those nearest neighbors are then evaluated using the uncoded sets with the PMK for a final decision for a point in the query pattern.

Each point in the query pattern is independently classified to a member of the database. In a final fusion[5,6] of decisions, we simply choose the most frequently occurring decision or mode.

5. EXPERIMENTAL RESULTS

Based on data sets described in Section 2, we applied invariant mapping and generated hash coded databases. Implementation of the classification algorithm requires setting the measurement error covariance matrix Σ and PMH parameters based on the physics of the application.

To design Σ in the Mahalanobis distance, we choose to populate only the main diagonal for simplicity. Our strategy begins by creating rough estimates for the measurement error variances of the five attributes. Suppose that we are 95% confident that our scattering center peak locations x and y are within two pixels of their true locations (i.e. within a 12 cm). Thus,

$$2 = 1.98\sigma_x \Rightarrow \sigma_x^2 = \sigma_y^2 = 1. \tag{10}$$

Now, suppose that the azimuth attribute for each peak is uniformly distributed over the 5 degree aperture; hence

$$\sigma_\theta^2 = \frac{(b-a)^2}{12} = \frac{5^2}{12} = 2. \tag{11}$$

For the amplitude attribute, log-normal and quarter-power normal models have been widely adopted for non-clutter peaks.[31] Under the log-normal assumption, suppose that with 95% confidence, the amplitude measurement is accurate within 5 dB (zero dB is the maximum peak in the data set); thus

$$5 = 1.98\sigma_A \Rightarrow \sigma_A^2 = 2.5^2 = 6.25. \tag{12}$$

Lastly, suppose that the even/odd bounce bit ν is correct with probability $p = 0.95$; thus

$$\sigma_\nu^2 = p(1-p) = 0.95(1-0.95) = 0.05. \tag{13}$$

For the invariant mapping, we choose to set $K = 32$ for the nearest neighbors in the local groups; a typical point pattern had approximately 200 points. After invariant mapping, all sets in $\mathcal{H}$ had 32 members each with the five dimensions of Equation (1). A pyramid histogram with a base level side of $N_0 = 512$ was sufficient to digitize all of the data; as a result, the histogram pyramids had $L = 9$ levels. The following sections apply these parameters to the CVDomes and 2006 collection experiments.

5.1 Classification Results using CVDomes

In addition to classification performance, this section shows the effect on performance of using hash codes. Recall, as described in Section 2.1, we are classifying vehicles placed on the red dots of Figure 7c based on a training vehicles placed on the circled dots.

Figure 7a shows a histogram classifying vehicle 5, sample 1 (a pose rotation at the center), using only 1% of the Hamming nearest neighbors and a 120-bit hash key. The histogram shows that 126 of 184 points in the attributed point pattern were properly classified in the database, and the mode is convincing. The misclassification errors are caused by quantization errors during the imaging, too few Hamming nearest neighbors, and measurement error. To make the problem more difficult, Figure 7b shows classification results for vehicle 5 imaged at the edge of the scene. The peak in the histogram is smaller but clear. The farthest point from a training sample is sample 44 labeled in Figure 7c; as shown in Figures 7d and 7e, using too few Hamming nearest neighbors causes a misclassification. Finally, in Figure 7f, we show an example using approximate nearest neighbors, where time complexity scales with the square root of the database size.

By applying the PMK to all 440 test samples, we achieved a peak classification performance of 98.7%. Depending on the hash key length and number of nearest neighbors in the Hamming distance, performance can degrade. Thus, the hashing provides a performance versus time-complexity trade-off. Figure 8 shows the effects of key length and the number of Hamming nearest neighbors on performance. As shown, a 120-bit key length using 400 Hamming nearest neighbors (3.4% of the database of 11875 hash codes) was sufficient for peak performance. The example confusion matrix in Figure 8 shows the methodology in calculating performance, where P_{id} is the trace divided by 440. Notice that most of the few errors are caused by distinguishing two model years of the Jeep, and all errors occurred for $q = 0.4$, i.e., in the outer ring of the scene.

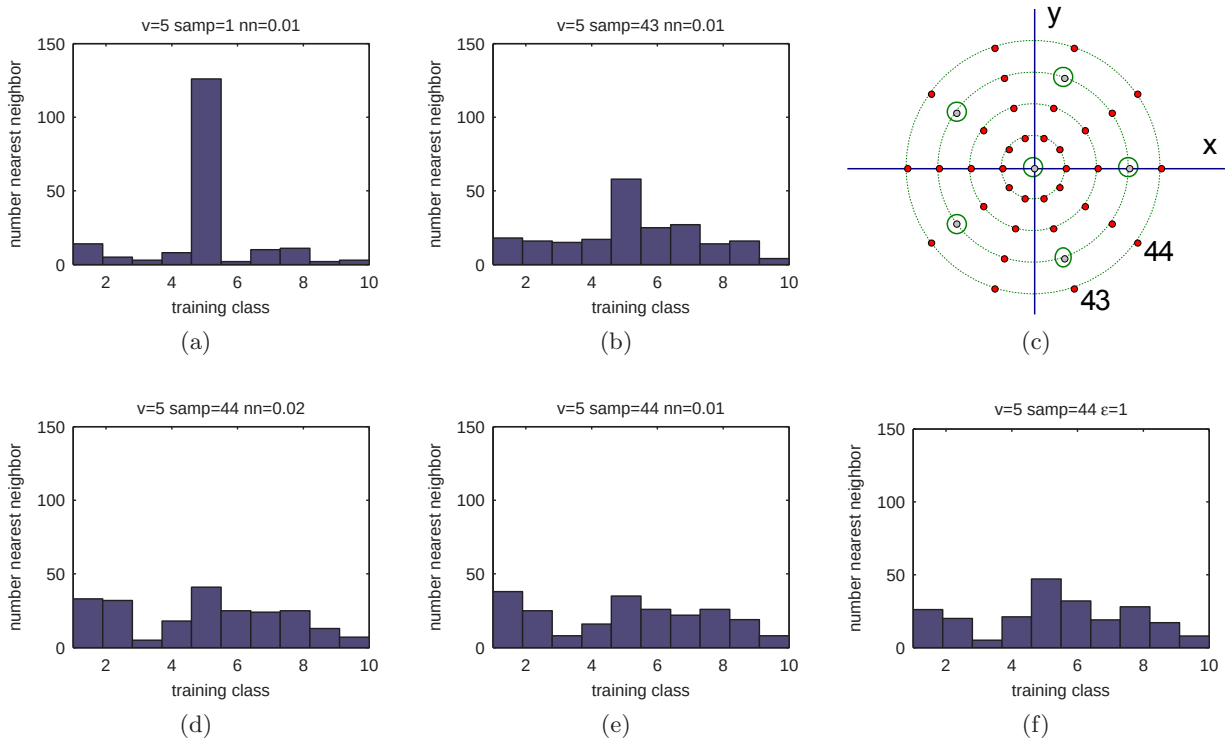
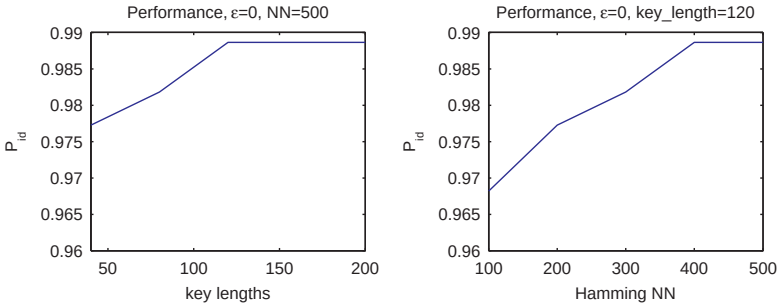


Figure 7. A query sample (vehicle) contains on the order of 150 scattering centers; each are identified to a member of the training set. A simple mode based fusion method identifies the query. In (a), v5 is rotated and placed at the center is easily identified. In (b), (d), (e), and (f), the vehicle is placed at the edge of the scene (c) and is more difficult to classify. Notice that setting the Hamming NN parameter too small causes a misclassification in (e).

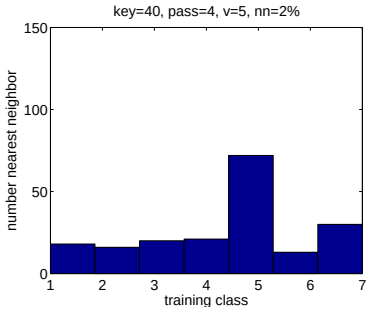


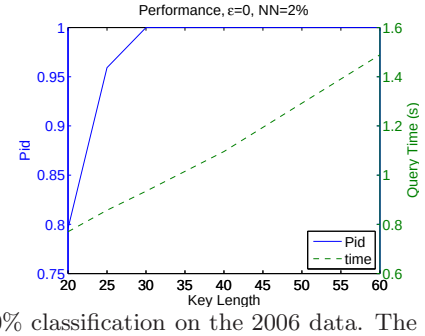
Actual vs. Classified, key=120, NN=500

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	44	0	0	0	0	0	0	0	0	0
2	0	44	0	0	0	0	0	0	0	0
3	0	1	40	3	0	0	0	0	0	0
4	0	0	0	44	0	0	0	0	0	0
5	0	0	0	0	44	0	0	0	0	0
6	0	0	0	0	0	44	0	0	0	0
7	0	0	0	0	0	0	43	0	1	0
8	0	0	0	0	0	0	0	44	0	0
9	0	0	0	0	0	0	0	0	44	0
10	0	0	0	0	0	0	0	0	0	44

Figure 8. Performance of the classifier on the CVDomes imagery varies with respect to hash key length and number of nearest neighbors in a Hamming space. The peak performance is achieved using greater than 120-bit keys and 400 Hamming distance nearest neighbors. The confusion matrix shows 98.7% performance in our application.

Using MATLAB with some mex C files, calculating the PMK required 13.4ms on a 2.67 GHz Intel Core i7. Thus, calculating the PMK for all 11875 members of the database would require 159 seconds. By calculating only the PMK on the 400 Hamming nearest neighbors, the required CPU time was just over 5 seconds. The overhead required to hash code the query and find the Hamming nearest neighbors required approximately 5 seconds. Hence, hash coding reduced query times from 159 s to 10 s. We expect improved coding and moving the algorithm to C would reduce query times significantly. We did not fully evaluate the performance of using approximate Hamming nearest neighbors, but this should also reduce time-complexity, especially on large databases.





Actual vs. Classified, key=30, NN=2%

	1	2	3	4	5	6	7
1	7	0	0	0	0	0	0
2	0	7	0	0	0	0	0
3	0	0	7	0	0	0	0
4	0	0	0	7	0	0	0
5	0	0	0	0	7	0	0
6	0	0	0	0	0	7	0
7	0	0	0	0	0	0	7

Figure 9. It is possible to achieve 100% classification on the 2006 data. The histogram shows an example of v5 correctly classified with a 40-bit hash key. Performance peaks with a 30-bit hash key and time complexity is directly proportional to key length. The confusion matrix shows all seven passes classified properly.

5.2 Classification Results using the Circular SAR 2006 Collection

The experiment for the 2006 collection was described in Section 2.2, where training samples are generated from the first pass of the SAR and test samples are generated from the remaining seven passes. The peak performance of classifying the vehicles using the PMK was 100%, however, we could evaluate performance versus reduced time complexity.

Figure 9 shows some results from applying PMH to the 2006 collection. The first plot is a histogram generated when classifying vehicle 5 on pass 4 using a 40-bit hash key and 2% of the Hamming nearest neighbors; vehicle 5 is clearly selected as the decision. Although not shown, the peak is more pronounced as the hash key length is increased to 80 or 120. The middle plot shows that a 30-bit hash code was sufficient to achieve 100% classification for the 49 test samples with approximately 0.9 second query times. The middle plot also shows that query times increase as hash key length increases. The confusion matrix is shown at the right of Figure 9.

6. DISSCUSION

Notice that the 2006 collection experiment has only 7 training samples versus the 60 training samples of the CVDomes experiment; both had approximately 200 attributed scattering centers per training sample. The smaller training set along with a shorter hash key explains the 10:1 ratio of query times. By using the uncoded PMK or a long hash key, we achieved 100% performance by a wide margin (i.e. the decision histogram had a strong peak for every test sample). The relative ease of identifying vehicles in the 2006 collection was due to the vehicles being place near the scene center and remaining stationary during the eight passes; any changes would be caused by noise, quantization errors, and focusing errors. This was our motivation for simulating the more challenging experiment using CVDomes.

Anecdotally, test samples that are spatially closer to a training sample (see Figure 7c) are easier to identify. This is due to the anisotropy of SAR imagery; vehicles look different from every azimuth and elevation. We found that it was adequate to place six training samples in a large scene when identifying well focused images. Other sources of error such clutter, occlusion, and defocus would reduce the range of certainty around training samples. Thus, we believe well focused data to a digital elevation map[32] may be important to identify civilian vehicle from X-band circular SAR in a single pass. We also identified vehicles based on 360 degrees of aspect around each vehicle; limited aspect can reduce classification performance significantly.[11]

Based on the two experiments in this paper, our algorithm applying invariant mapping along with PMH is successful at identifying civilian vehicles from circular SAR imagery in near real-time. The algorithm is fast, scalable to large databases, and expandable to additional information such as 3D or other scattering attributes.

ACKNOWLEDGMENTS

This material is based upon work supported by the Air Force Research Laboratory under Award FA8650-07-D-1220 and by the Air Force Office of Scientific Research under Award FA9550-06-1-0324. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily

reflect the views of the U.S. Air Force. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

REFERENCES

[1] Potter, L. C. and Moses, R. L., "Attributed scattering centers for SAR ATR," *IEEE Transactions on Image Processing* **6**(1), 79–91 (1997).

[2] Tico, M. and Kuosmanen, P., "Fingerprint matching using an orientation-based minutia descriptor," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(8) (2003).

[3] Jain, A. K., Feng, J., Nagar, A., and Nandakumar, K., "On matching latent fingerprints," in [*IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*], 1–8 (June 23-28 2008).

[4] Grauman, K. and Darrell, T., "Pyramid match hashing: sub-linear time indexing over partial correspondences," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (June 2007).

[5] Kittler, J., Htef, M., Duin, R., and Matas, J., "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(3) (1998).

[6] Kuncheva, L. I., "A theoretical study on six classifier fusion strategies," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(2) (2002).

[7] Grauman, K. and Darrell, T., "The pyramid match kernel: efficient learning with sets of features," *Journal of Machine Learning Research* **8** (2007).

[8] Dungan, K. E. and Potter, L. C., "Classifying civilian vehicles using a wide-field circular SAR," in [*Algorithms for Synthetic Aperture Radar Imagery XVI*], Zelnio, E. G. and Garber, F. D., eds., *Proc. SPIE* **7337**, 73370R (Apr. 28 2009).

[9] Dungan, K. E., Austin, C., Nehrbass, J., and Potter, L. C., "Civilian vehicle radar data domes," in [*Algorithms for Synthetic Aperture Radar Imagery XVII*], Zelnio, E. G. and Garber, F. D., eds., *Proc. SPIE* **7699** (2010).

[10] Casteel, Jr., C. H., Gorham, L. A., Minardi, M. J., Scarborough, S. M., Naidu, K. D., and Majumder, U. K., "A challenge problem for 2D/3D imaging of targets from a volumetric data set in an urban environment," in [*Algorithms for Synthetic Aperture Radar Imagery XIV*], Zelnio, E. G. and Garber, F. D., eds., *Proc. SPIE* **6568**, 65680D (2007).

[11] Dungan, K. E., Potter, L. C., Blackaby, J., and Nehrbass, J., "Discrimination of civilian vehicles using wide-angle SAR," in [*Algorithms for Synthetic Aperture Radar Imagery XV*], Zelnio, E. G. and Garber, F. D., eds., *Proc. SPIE* **6970**, 69700Z (2008).

[12] Sim, D.-G., Kwon, O.-K., and Park, R.-H., "Object matching algorithms using robust Hausdorff distance measures," *IEEE Transactions on Image Processing* **8**(3), 425–429 (1999).

[13] Jakowatz, Jr., C. V., Wahl, D. E., Eichel, P. H., Ghiglia, D. C., and Thompson, P. A., [*Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*], Kluwer Academic Publishers, Norwell, MA (1996).

[14] Gorham, L. A., Majumder, U. K., Buxa, P., Backues, M. J., and Lindgren, A. C., "Implimentation and analysis of a fast backprojection algorithm," in [*Algorithms for Synthetic Aperture Radar Imagery XIII*], Zelnio, E. G. and Garber, F. D., eds., *Proc. SPIE* **6237**, 62370G (May 17 2006).

[15] Wahl, D. E., Yocky, D. A., and Jakowatz, Jr., C. V., "An implementation of a fast backprojection image formation algorithm for spotlight-mode sar," in [*Algorithms for Synthetic Aperture Radar Imagery XV*], Zelnio, E. G. and Garber, F. D., eds., *Proc. SPIE* **6970**, 69700H (Apr. 15 2008).

[16] Dungan, K. E. and Potter, L. C., "Three-dimensional imaging of vehicles with wide-angle synthetic aperture radar," *IEEE Trans. Aerosp. Electron. Syst.* (accepted for publication).

[17] Giuli, D., "Polarization diversity in radars," *Proceedings of the IEEE* **74**, 245–269 (Feb. 1986).

[18] Novak, L. M., Burl, M. C., and Irving, W. W., "Optimal polarimetric processing for enhanced target detection," *IEEE Trans. Aerosp. Electron. Syst.* **29**, 234–244 (Jan. 1993).

[19] Ertin, E. and Potter, L. C., "Polarimetric classification of scattering centers using M-ary Bayesian decision rules," *IEEE Transactions on Aerospace and Electronic Systems* **36**, 738–749 (July 2000).

[20] Keller, J. B., "Geometrical theory of diffraction," *J. Opt. Soc. Amer.* **52**(3), 116–130 (1962).

[21] Ettinger, G. J., Klanderman, G. A., Wells, W. M., and Grimson, W. E. L., "Probabilistic optimization approach to SAR feature matching," in [*Algorithms for Synthetic Aperture Radar Imagery III*], Zelnio, E. G. and Douglass, R. J., eds., *Proc. SPIE* **2757**, 318–329 (June 10 1996).

[22] Chiang, H.-C., Moses, R. L., and Potter, L. C., "Model-based classification of radar images," *IEEE Transactions on Information Theory* **46**, 1842–1854 (Aug. 2000).

[23] Chiang, H.-C., Moses, R. L., and Potter, L. C., "Model-based Bayesian feature matching with application to synthetic aperture radar target recognition," *Pattern Recognition* **34**(8), 1539–1553 (2001).

[24] Bhanu, B. and Jones III, G., "Increasing the discrimination of synthetic aperture radar recognition models," *Opt. Eng.* **41(12)**, 3298–3306 (2002).

[25] Bhanu, B., Jones III, G., and Wang, R., "Composite class models for SAR recognition," in [*Algorithms for Synthetic Aperture Radar Imagery X*], Zelnio, E. G. and Garber, F. D., eds., *Proc. SPIE* **5095**, 284–291 (Sept. 12 2003).

[26] Irving, W. W., Washburn, R. B., and Grimson, W. E. L., "Bounding performance of peak-based target detectors," in [*Algorithms for Synthetic Aperture Radar Imagery IV*], Zelnio, E. G., ed., *Proc. SPIE* **3070**, 245–257 (July 28 1997).

[27] Brendel, G. G. and Horowitz, L. L., "Benefits of aspect diversity for SAR ATR: fundamental and experimental results," in [*Algorithms for Synthetic Aperture Radar Imagery VII*], Zelnio, E. G., ed., *Proc. SPIE* **4053**, 567–578 (Aug. 24 2000).

[28] Ettinger, G. and Snyder, W., "Model-based fusion of multi-look SAR for ATR," in [*Algorithms for Synthetic Aperture Radar Imagery IX*], Zelnio, E. G., ed., *Proc. SPIE* **4727**, 277–289 (Aug. 1 2002).

[29] Bhanu, B. and Jones III, G., "Exploiting azimuthal variance of scatterers for multiple look SAR recognition," in [*Algorithms for Synthetic Aperture Radar Imagery IX*], Zelnio, E. G., ed., *Proc. SPIE* **4727**, 290–298 (Aug. 1 2002).

[30] Saito, M. and Matsumoto, M., "SIMD-oriented fast Mersenne Twister: a 128-bit pseudorandom number generator," in [*Monte Carlo and Quasi-Monte Carlo Methods 2006*], 607–622, Springer (2008).

[31] DeVore, M. D. and O'Sullivan, J. A., "Statistical assessment of model fit for synthetic aperture radar data," in [*Algorithms for Synthetic Aperture Radar Imagery VIII*], Zelnio, E. G., ed., *Proc. SPIE* **4382**, 358–366 (Aug.27 2001).

[32] Jakowatz, Jr., C. V., Wahl, D. E., and Yocky, D. A., "Beamforming as a foundation for spotlight-mode SAR image formation by backprojection," in [*Algorithms for Synthetic Aperture Radar Imagery XV*], Zelnio, E. G. and Garber, F. D., eds., *Proc. SPIE* **6970**, 69700Q (Apr. 15 2008).