Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Signal Processing 84 (2004) 785–799

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Detection of linear objects in GPR data

Andrea Dell'Acqua[a], Augusto Sarti[a,*], Stefano Tubaro[a], Luigi Zanzi[b]

[a] *Dip. di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo Da Vinci 32, 20133 Milano, Italy*
[b] *Dip. di Ingegneria Strutturale, Politecnico di Milano, Piazza Leonardo Da Vinci 32, 20133 Milano, Italy*

Received 25 October 2002; received in revised form 9 October 2003

Abstract

In this paper, we propose a semi-automatic approach to the detection of linear scattering objects in geo-radar data sets, based on the 3D radon transform. The method that we propose is iterative, as each detected object is removed from the data set before the next iteration, in order to avoid mutual interference or masking. In addition, the algorithm is able to further analyze the data set in a local fashion in order to eliminate spurious targets from the set of lines of maximum consensus.

Our algorithm proved robust and reliable even in the presence of data affected by heavy noise, artifacts and other undesired scattering objects.

Although the application scenario of the proposed algorithm is that of the analysis of data sets generated by a ground penetrating radar, the method is general enough to apply to any problems where linear objects needs to be identified and localized in volumetric data.

Keywords: Radon transform; Hough transform; Line detection

1. Introduction

Detecting, localizing and assessing buried or hidden objects in a non-destructive and cost-effective fashion is a problem of great interest for a variety of application areas that range from structural engineering to medical and defense.

The ground penetrating radar (GPR) is widely considered an excellent non-destructive method for the detection of underground objects such as pipes, beams, tunnels, buried walls, etc. Typical data are 2D vertical sections (slices) of the ground, where hyperbolic patterns reveal the presence of scattered objects. The detection of targeted objects in such data, however, is often left to a human operator, due to the presence of intense noise, artifacts, and interfering objects (e.g. stones). Nonetheless, many attempts to automatize this pattern recognition task have been proposed in the literature (see, for example, [6,9,27]), but they are usually limited to a 2D domain (volume slices) and, therefore, they do not take advantage of the interdependence between aligned views. In order to overcome this limitation it is thus necessary to work directly on the 3D data set obtained by packing together all the available parallel slices. In general, this is accomplished by first applying a

* Corresponding author. Tel.: +39-2-2399-3647; fax: +39-2-2399-3413.
E-mail address: augusto.sarti@polimi.it (A. Sarti).

0165-1684/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2003.12.010

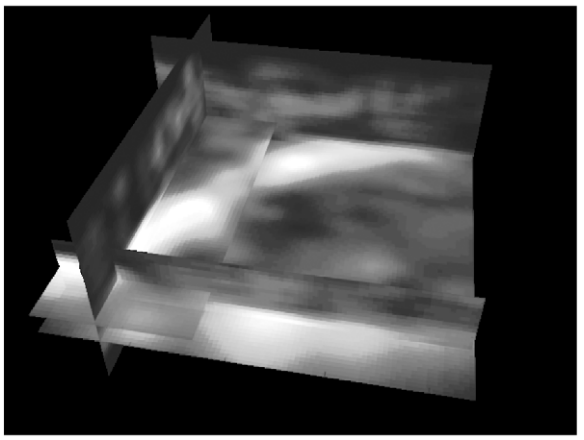

Fig. 1. An assembly of slices of a real 3D data set after applying the migration operator. These data correspond to prospections of the ground underneath a gas station in Gaggiano, Italy.

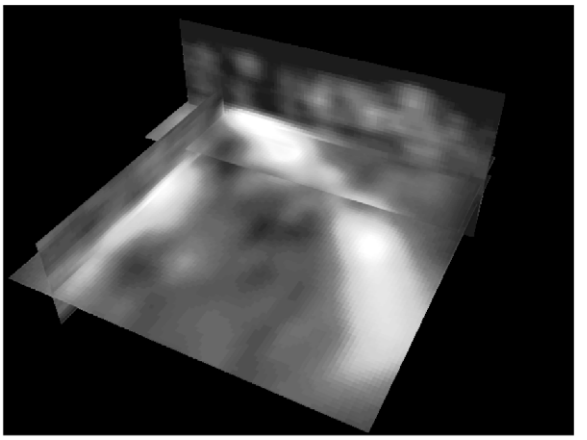

Fig. 2. A different view of the "Gaggiano" data set.

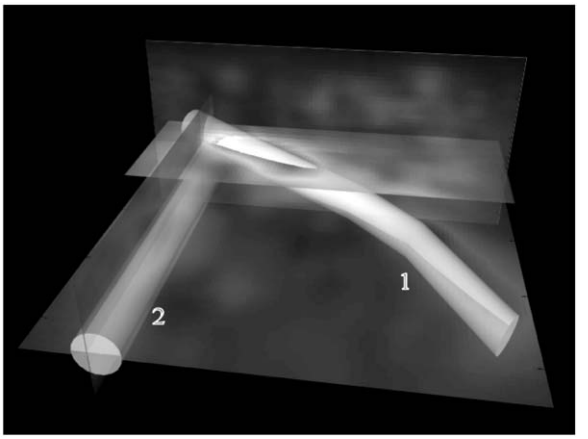

Fig. 3. Actual location of buried pipes in the "Gaggiano" data set.

migration [1] operator to the 3D data set, in order to obtain a data set that is able to describe the object shapes in a metrically consistent fashion. This approach is particularly convenient with linear target objects, in which case the hyperbolic surfaces generated by such targets collapse into roughly cylindrical shapes that approximate the objects to be detected. This helps the user distinguish the real targets from other non-elongated interfering objects (see Figs. 1–3).

Detection of linear objects is a problem of great interest, particularly for application of industrial/civil engineering (localization of pipes, metal rods, etc.). In the past decade, methods for linear object detection have appeared in the remote sensing literature for 2D data (see, for example, road detection techniques in SAR [18] or LADAR [25] images. Such line detection techniques, however, have little use for 3D data obtained with GPR, which looks more like a noisy collection of aligned elongated spots (see Figs. 1 and 2). In fact, the lack of continuity and the non-negligible thickness of the "blobs" that constitute the data relative to each target line are likely to cause serious difficulties to traditional line detection methods based on local data analysis. In order to overcome these problems we must turn to solutions based on a more "global" approach.

Examples of global analysis tools that seem appropriate for the data that we are interested in are *Hough* and *Radon* transforms. There is a twofold reason why such transforms can provide us with a viable approach to the solution of the faced problem. The former is their intrinsic robustness against data discontinuities. In fact, they could help detect linear targets even if they appear as a series of aligned but isolated blobs. The latter reason that motivates our interest in such

[1] A migration is an inversion operation involving rearrangement of seismic information elements so that reflections and diffractions are plotted at their true location [23, p. 193]. When a data set undergoes migration, the energy of a scattering "spot" in the ground, formerly distributed along the diffraction hyperbola by the acquisition processing, collapses in the original spot.

transforms is their intrinsic robustness against the impact of undesired scattering objects such as stones.

The Hough transform (HT) was originally conceived as a tool for detecting lines in 2D images, but numerous extensions of this algorithm can be found in the literature for the detection of more general shapes that belong to specific parametric families, such as ellipses [1,17]. Further results are also available for the detection of non-parametric shapes (see generalized HT [4,8,16]). Unfortunately, computational cost and memory requirements make HT-based solutions extremely sensitive to the number of parameters that are needed to model the shapes to be detected. For these reasons, the literature is rich with techniques aimed at the optimization of the basic algorithms (see fast linear HT [13], iterative HT [16], adaptive HT [12], fast adaptive HT [10], randomized fuzzy HT [17], etc.). An interesting example of this sort is presented in [7] and [3], which propose a hierarchical and multi-resolution approach to HT analysis.

An example of application of a 3D HT to the analysis of GPR data sets was recently proposed in [2].

In this work we propose a solution based on the 3D radon transform (RT), which has been shown to be equivalent to the Hough Transform for the detection of lines [5,24]. It is well known that the RT is commonly used for the generation of 3D tomographic data from 3D slices [14,26], but it has also been used for detection of lines in both 2D and 3D spaces [15,19].

In the next section we will provide the basics on the Radon transform and the extension of the RT to the 3D domain. Such concepts will then be applied to the analysis of 3D data sets. We will also provide criteria to correctly interpret the RT of complex data (see Section 3). The proposed solutions will be tested on both simulated and real data (see Section 4).

2. Basics on parametric line detection

2.1. 2D radon transform

Let $f(x, y)$ be an image and $g(\gamma, u)$ its 1D projection along the direction γ

$$g(\gamma, u) = \int_{l_{\gamma,u}} f(x, y)\, \mathrm{d}l.$$

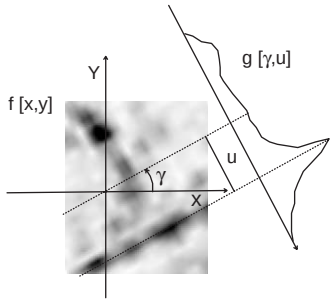

Fig. 4. An image $f(x, y)$ (in inverse video scale) and its projection $g(\gamma, u)$ along the direction γ.

According to this definition, $g(\gamma, u)$ is the line integral of the image intensity, computed along a line l whose distance from the origin is u and whose angle from the X-axis is γ (Fig. 4). All points (X, Y) on the line l satisfy the equation

$$u = X \sin(\gamma) - Y \cos(\gamma)$$

and this allows us to rewrite the projection function $g(\gamma, u)$ as

$$g(\gamma, u) = \int f(x, y)\delta(x \sin(\gamma) - y \cos(\gamma) - u)\, \mathrm{d}x\, \mathrm{d}y,$$

where $\delta(\cdot)$ is the Dirac function. The family of functions of the form $g(\gamma, u)$ for all γ is called Radon transform (RT) of the image $f(x, y)$.

In practice, the basic RT algorithm considers one cell of the transform at a time and sums together pixel values in data-space along the corresponding line. This, however, is not the only way to compute the RT: more efficient methods have, in fact, been developed. Examples of this sort are the solutions that work in the frequency domain [19], which are based on the Fourier slice theorem (see, for example, [14]).

2.2. Parametrization of 3D lines

There are several ways to describe a line in a 3D space. In a coordinate frame (X, Y, Z), a line that passes through the point (x_0, y_0, z_0) can be readily expressed in a *parametric* form

$$\begin{aligned} x &= x_0 + at, \\ y &= y_0 + bt, \\ z &= z_0 + ct, \end{aligned} \tag{1}$$

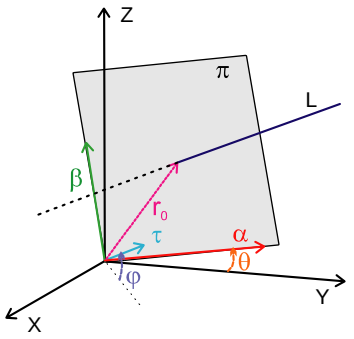

Fig. 5. The orthogonal basis τ, α, β. θ is the angle between α (azimuth) (lying on the XY plane) and the Y-axis. φ is the elevation angle of τ (with respect to XY). r_0 is the offset vector of the line L, and lies on the π plane.

where a, b, c represent the components of the line's direction along the three reference axes. Of course, the parameter t decides which point (x, y, z) of the line we are considering.

A simple rearrangement of these three equations leads to the so-called *symmetric* form

$$\frac{x - x_0}{a} = \frac{y - y_0}{b} = \frac{z - z_0}{c}.$$

One drawback of this representation, however, is its redundancy. In fact, we know that a line can be completely specified by four parameters. In order to come to a minimal representation of lines, we begin by rewriting Eq. (1) [26] as

$$\mathbf{r} = \mathbf{r}_0 + t\boldsymbol{\tau},$$

which gives the generic point $\mathbf{r}$ on the line as a displacement from the starting point $\mathbf{r}_0$ (offset vector) along the direction $\boldsymbol{\tau}(|\boldsymbol{\tau}| = 1)$ of an amount set by the parameter t. Let π be a plane that is perpendicular to $\boldsymbol{\tau}$ and passes through $\mathbf{r}_0$. If $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are unit vectors that span this plane, then $\mathbf{r}_0$ can be rewritten as

$$\mathbf{r}_0 = u_0\boldsymbol{\alpha} + v_0\boldsymbol{\beta}.$$

We can choose $\boldsymbol{\tau}, \boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ to form an orthogonal basis (Fig. 5) with $\boldsymbol{\alpha}$ lying on the (X, Y) plane. It is possible to describe the orientation of a line L with respect to the new (X, Y, Z) frame using the angles θ and φ, where θ is the angle between $\boldsymbol{\alpha}$ and the Y-axis (*azimuth* of the line measured on the XY-plane) and φ is the angle that describes the *elevation* of $\boldsymbol{\tau}$ from the XY-plane.

Using this parameterization we obtain

$$\boldsymbol{\tau} = \begin{pmatrix} \cos(\theta)\cos(\varphi) \\ \sin(\theta)\cos(\varphi) \\ \sin(\varphi) \end{pmatrix}, \boldsymbol{\alpha} = \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \\ 0 \end{pmatrix},$$

$$\boldsymbol{\beta} = \begin{pmatrix} -\cos(\theta)\sin(\varphi) \\ -\sin(\theta)\sin(\varphi) \\ \cos(\varphi) \end{pmatrix},$$

therefore a line is described by just four parameters: two for the orientation (θ, φ), and two (u_0, v_0) for the offset from the origin on the XY plane.

In order to exhaustively analyze the data set, the azimuthal range will be $0 \leqslant \theta < \pi$, while the elevation range will be[2] $-\pi/2 \leqslant \varphi < \pi/2$.

2.3. Detection of linear volumetric elements

When looking at a real geo-radar data set (see Fig. 6), the detection of linear scattering elements (pipes, beams) appears to be very complex, principally for two reasons:

1. target elements have a non-negligible thickness (they are more than one voxel thick), which may vary in shape (usually they are cylindrical or parallelepipedal);
2. scattered images of the target elements are not continuous, but they often reduce to a collection of elongated blobs. (An example of this sort is shown in Fig. 7, which is a vertical section of the data set of Figs. 1, 2 and 3).

For these reasons it is hard to interpret the data provided by an RT, with the consequence that a simple peak-detection algorithm in the transformed space is likely to fail.

[2] In order to speed up the algorithm, we can restrict the range of φ depending on the specific application. For example, in order to detect pipes buried in the ground it is usually sufficient to choose $-\pi/4 \leqslant \varphi < \pi/4$. In fact, this way the pipes with high slope will not be detected, but we know that such pipes are usually connected with detectable horizontal pipes, therefore all the areas where digging could likely damage the pipes will be easily found.

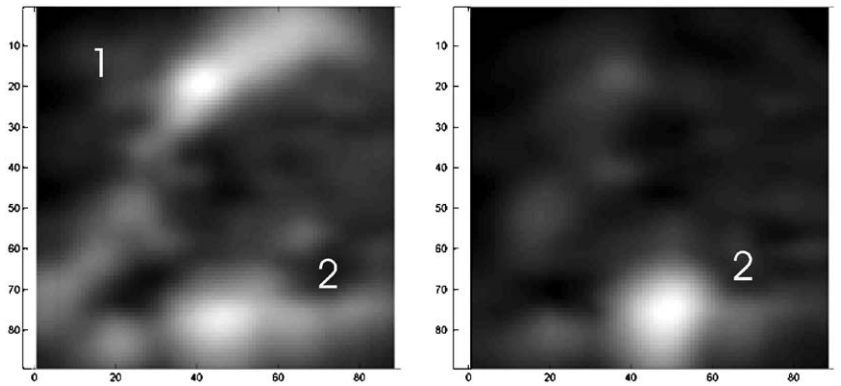

Fig. 6. Two horizontal sections of the data set 'Gaggiano' taken at two different depths. There are two pipes (see Figs. 1, 2, also 3). The pipe no. 2 does not keep its original elongated shape.

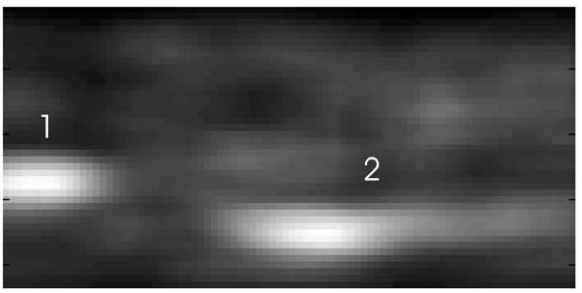

Fig. 7. Vertical section of the data set 'Gaggiano'. The left spot is the section of a pipe (1) perpendicular to the image plane. The right one (2) is the longitudinal section of another pipe. The pipes are both visible in Fig. 3 and from a top view in Fig. 6.

Due to the non-negligible thickness of the scattering targets, there is a region, around each peak in the parameter space, where the cumulation array takes on significant values, corresponding to all the lines that pass through the object approximately along its linear extension. In fact, the line of "maximum consensus" (which is the line of integration that returns the highest value) does not necessarily lie along the target's principal axis, as that may not be the direction of maximum consensus. For example, the maximum consensus in the case of the object of Fig. 8 is achieved along its diagonals.

In order to overcome these problems, we can adopt different strategies. A good theoretic approach could be to implement an HT in such a way to detect cylindrical objects. This, however, would force us to deal with a 5-parameter space (we would need an additional parameter to encode the diameter of the cylinder), with dramatic consequences on the computational complexity and on the memory requirements. Another solution is to integrate along "fat rays" [11], which are bundles of parallel lines, and sum the contribution of all rays in the bundle. The bundle thickness and the sampling step are, indeed, closely related, as they both should roughly coincide with the thickness of the pipes in the data set, which can be estimated through visual inspection of the volumetric data.

The approach we chose consists of integrating along tightly packed one-voxel-wide lines (with a sampling step that is approximately equal to one voxel), and then filtering each projection plane with a 2D mask whose size corresponds, once again, to the thickness of the pipes. This solution is typically adopted in RT-based applications, such as tomographic ones [14], and is equivalent to the above-mentioned others from the computational complexity standpoint.

In order to correctly identify the orientation and the position of a pattern of non-negligible thickness, a good solution is to first identify the pattern as if it had no thickness (a line, a plane) and afterwards to find how many voxels belong to the detected structure [21]. The axis of a cylindrical target, for example, can be found through a regression process based on the minimization of the mean square distance from the unknown axis of the voxels that has been classified as belonging to the target (see next section)

$$L = \sum_{i=1}^{N} l_i^2,$$

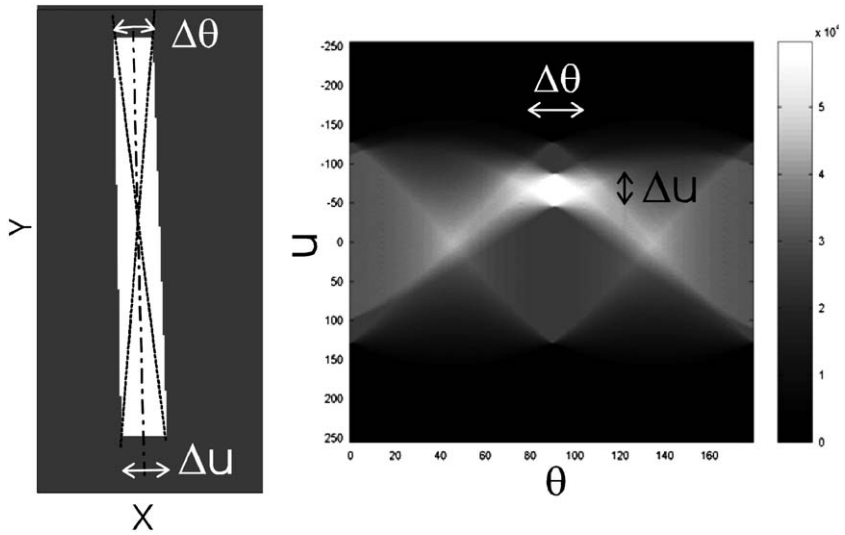

Fig. 8. A 2D image with a rectangle (left) and its RT (right). The accumulation array takes on significant values in a diamond-like region with diagonals Δu (offset range) and $\Delta\vartheta$ (angle range).

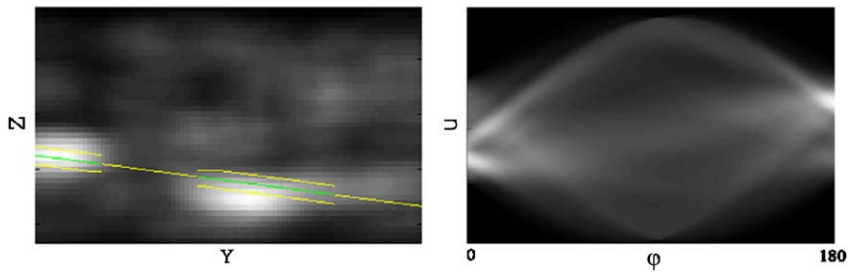

Fig. 9. On the left: a vertical section of the data set "Gaggiano" (Fig. 7). On the right: its 2D RT. The line of "maximum consensus" is at $\varphi = 170°$, but it does not correspond to a real target, because the two spots belong to different pipes (see Fig. 3).

where N is the total number of target voxels and l_i is the distance of the center of the ith voxel from the cylinder axis.

When there is more than one target in the data set, another problem arises: clouds of high-valued points (one per target) often overlap in the projection space. This tends to modify the geometry of overlapped regions, and very often stronger cumulation regions tend to "mask" weaker ones. Another, often more serious, problem is the discontinuity in the shape of pipes, which often results in the detection of spurious pipes or beams through peak analysis of the RT. In fact high-value regions may be generated through integration along lines that pass through different real objects.

An example of spurious target detection is shown in Fig. 9. In the next section, we will address all such problems and show a strategy to achieve a reliable target detection.

3. An effective approach to linear object detection

3.1. An iterative approach

Due to the presence of overlapping regions in the parameter space, the simultaneous identification of more than one scattering object is often too much of a complex problem. Furthermore, peaks in parameter

space are often generated by lines that pass through "blobs" of various scattering objects. In order to be able to cope with these problems, we can make the process iterative and detect only the leading target at each step. In fact, once the most likely target is detected, it can be removed from the data set by setting to zero all the voxels that belong to the target. The next iteration will be performed on the "residual" data set (what is left after target removal).

This method tends to improve estimation accuracy and prevent spurious target detection. In the section on 'Experimental results' we will show an example of application of this method.

3.2. Target detection in complex data

The first step of each iteration is to detect all the significant maxima in the RT domain. After that, we need to select the most likely target and subtract it from the data set. In order to characterize the likelihood of every potential target and guide the selection of the best candidate, we need to define some figure of merit. As already said above, the target's likelihood is rather difficult to assess just by inspecting the RT, while more information can be found in the data space.

From Figs. 1, 2, 6 and 7 we immediately notice that a target tends to look like a noisy collection of aligned spots or "blobs". In order to verify whether a set of blobs belong to a single target, we propose a validation criteria based on the evaluation of two parameters: the *relative position* and the *alignment* with the axis of the target. Targets (maxima in RT) that do not correspond to real scatterers (whose blobs are generated by different elements) are generally characterized by blobs that are well separated one from each other and are elongated on a direction that does not correspond to the axis of the target, as each one tends to be aligned with the real target that it belongs to.

In order to quantify these features, we first need to locate the blobs that correspond to the target. This can be done by analyzing the value of the voxels along the *target line*, which is the line that corresponds to the selected peak in the RT domain. The intensity values of the voxels can be represented as a 1D function like the ones shown in Fig. 10. We can define a *spot* as a segment of the target line where the intensity keeps itself above a certain threshold (set for example to 50% of the peak value), or, in a more sophisticated way,

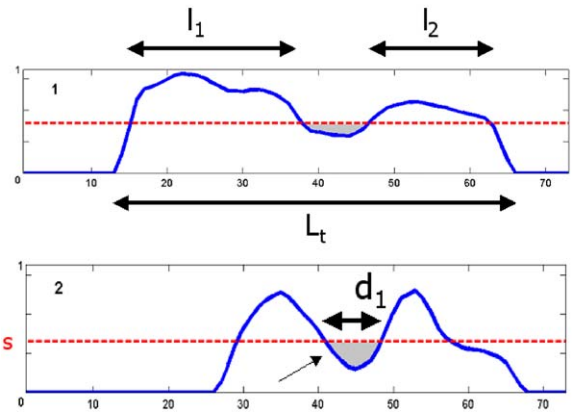

Fig. 10. Magnitude profile of the data set along two target lines. The threshold s (dashed line) is set to 50% of the profile's peak value. This allows us to distinguish the spots (both targets generate two spots). The H coefficient of each target is the area of the gray region between the spots.

we can use cluster detection techniques to distinguish the spots: a 1D spot can be seen as a 1D section along the target's axis of a 3D blob.

Once the blob length is determined, we need to quantify its thickness. In order to be able to remove a linear object and facilitate the detection of other objects, we need to determine its volume occupancy through localization of its side border. Unfortunately, the complexity of the data makes this problem quite challenging. In order to overcome this difficulty, we propose a rather simple and fast solution that proved effective in all the experiments that we conducted on real data.

Given the target line, we first decide in which direction to slice the volume (XY, YZ, ZX planes), in such a way that the slices are as orthogonal as possible to the line itself. We then determine slice by slice the region where the blob is. Given a voxel V on the target line, we start from its location on the slice and construct the blob area using a region growing approach. Neighbours are, in fact, visited and thresholded according to a pre-selected proximity mask. If a neighbour qualifies as belonging to the blob, its location is added to the region. This process guarantees that the blob points will satisfy criteria of connectedness and relevance. With the experiments that we conducted, we obtained good results by choosing a proximity mask that includes the four closest neighbours, and setting the threshold to

50% of the peak value. This process of 'expansion', however, is limited to a pre-assigned radius, which can be decided using some a priori knowledge on the data set and on the sampling step. Usually this radius is of a few voxels. This choice prevents the blob region from growing too much, perhaps by including areas that belong to another target nearby. Notice that, on the other hand, underestimating the blob size has no dramatic consequences, as the ignored portions would be later detected as a new target that turns out to be almost parallel and very close to the first one. It would thus be quite easy to merge them back together.

Working slice by slice has the advantage of speed, but ignoring correlations between adjacent slices could give rise to problems. In order to avoid such problems without giving up computational efficiency, our algorithm also includes an additional refining process that tends to eliminate undesired planar region that turns out to be inconsistent in subsequent slices.

Indeed, more sophisticated solutions can be devised in order to identify the actual volume occupancy of the linear objects. The solution that we propose, however, has the advantage of simplicity, and produces results that are reasonably good.

In order to evaluate the relative position of the blobs, it can be sufficient to analyze the 1D intensity profile. In particular, the parameter L can be defined as

$$L = \frac{\sum_{i=1}^{N_s} l_i}{L_t}, \tag{2}$$

where the target length L_t is the distance from the first voxel of the first spot to the last voxel of the last spot, while l_i is the length of the ith of the N_s spots. For values of L near to one the target is composed by a single blob or of blobs very close one to each other, therefore the target is probably a real one.

The relative position of spots can also be characterized by the behaviour of the intensity of the data between two spots. To quantify this behaviour, we can use

$$H = \frac{\sum_{i=1}^{N_s-1} \sum_{p=1}^{d_i} s - I(p)}{\sum_{k=1}^{N_s-1} d_k}, \tag{3}$$

where p is a point on the target line, $I(p)$ is the intensity value in p, s is a fixed threshold, and d_k is the length of the kth space between two adjacent spots (see Fig. 10). Therefore, the denominator of Eq. (3) is the total length of the $N_s - 1$ tracts of the target line that lie between two spots. The parameter H is thus the equivalent height of the gray area in Fig. 10.

A very important feature that is important to retrieve in our attempt to eliminate spurious targets is the *orientation* of a blob. In fact, blobs tend to keep the same orientation as the scattering object that generated them. In fact, a misalignment between the blobs and the direction of the target line is usually a sign of blob mismatch, which indicates that the blobs are likely to belong to different scattering objects.

The 3D orientation of a blob can be computed from the eigenvalues of the inertia tensor [20,22] of the blob itself. An eigenvalue represents the dispersion of the mass around the corresponding eigenvector, so the eigenvector related to the smallest eigenvalue is aligned with the principal axis of the blob. The scalar product between this eigenvector and the unit vector of the target line is a measure of the target–blob alignment (see Fig. 14).

Due to the strong noise that affects the real data set, some blobs could lose their elongated shape. In this case we cannot locate a principal axis (the blob is discoidal or spherical), therefore computing the scalar product between the target direction and eigenvector is no longer meaningful. This situation, however, can be easily known through direct inspection of the eigenvalues of the inertia tensor. In fact, when the smallest and the median eigenvalues are comparable, the "mass" is equally distributed along the corresponding eigenvectors. In conclusion, it should be quite clear that the scalar product between target direction and the eigenvector needs to be weighed with the uncertainty on the blob direction's estimate. Since it is difficult to correctly estimate the orientation of small blobs, we also need to weigh the information on every single blob's orientation with the length of the blob itself (that can be approximated by the length of the corresponding spot on the target line).

All the alignment information can be summarized by the coefficient

$$A = \sum_{m=1}^{N_s} \sigma_1(a)\sigma_2(g)p_m, \tag{4}$$

where a is the scalar product between target direction and eigenvector (see above), g is the reliability of the alignment, which is measured as the ratio ($g = \lambda_3/\lambda_2$) between the smallest (λ_3) and the median (λ_2)

eigenvalues of the inertia tensor, p_m is the weight of the mth blob, which is computed as

$$p_m = \frac{L_m}{\sum_{i=1}^{N_s} L_i},$$

where L_i is the length of the ith spot, and N_s is the total number of spots, σ_1 and σ_2 are two sigmoids, that we introduced after performing some test on real data sets in order to improve the reliability of A.

σ_1 is a sigmoid of the form

$$\sigma_1(x) = \frac{1}{1 + \exp(-k_1(x - \bar{x}_1))}, \quad (5)$$

which is a monotonically increasing function whose values range between 0 and 1. The transition from 0 to 1 occurs in a region that is centered in $x = \bar{x}$. This nonlinearity is introduced in order to avoid penalizing A for small misalignments, as pipes are often not perfectly linear (see Fig. 14). We chose $k = 15$ and $\bar{x} = 0.7$ in order to allow changes in the orientation of the targets smaller than 30° and strongly penalize misalignments that are worse than 45°.

In addition, from our tests we found that it is better to assign a fairly constant weight to all the blobs whose eigenvalues ratio g falls below a pre-defined threshold. For this reason, we introduced a decreasing sigmoid σ_2 of the form

$$\sigma_2(x) = 1 - \frac{1}{1 + \exp(-k_2(x - \bar{x}_2))}. \quad (6)$$

Experimental tests on synthetic and real data showed that choosing $k_2 = k_1$ and $\bar{x}_2 = \bar{x}_1$ produces good results.

3.3. Termination criterion

One issue to address on the above-discussed detection process is how to automatically terminate the algorithm's iterations. In fact, we know that the algorithm should continue its iterations until all the significant targets are identified, therefore it is necessary to identify a criterion for classifying a target as significant or irrelevant. One way to do so, could be to check its density, i.e. the mean value of the voxels along and around its axis. Quite obviously, when computing the mean value we should not consider all the voxels along the target line, but only those that belong to a spot. This would prevent us from penalizing those targets that cover only a portion of the volume (e.g. the vertical one in Fig. 11), or those that are split into separated spots due to noise.

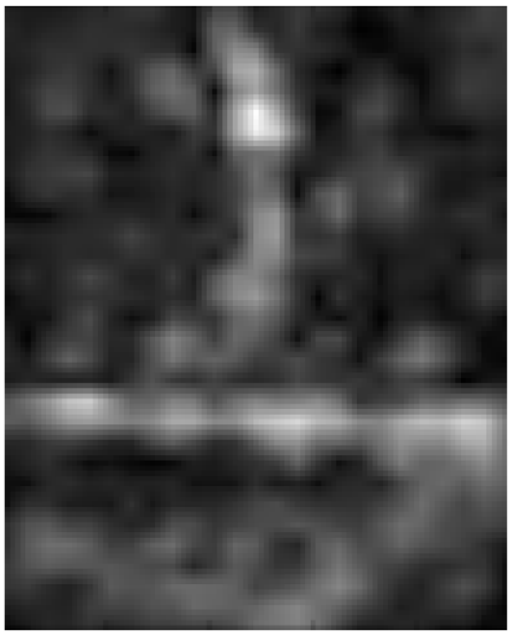

Fig. 11. T-junction of two pipes.

When the density of the last detected target is much lower than that of the first identified one, the iterative process can stop (Fig. 13).

Another criterion for terminating the iterations can be derived from the displacement of the local maxima in Radon space.

A situation that we should be able to handle is that of a data set without any linear target. In fact, aside from the case in which the data set is "empty", any data set is always reduced to an "empty" one after a number of iterations (iterative "removal" of pipes). The RT of an "empty" volume does not exhibit any significant peaks, and it looks like a "hilly" landscape with a large number of mild local maxima. In geo-radar data set such maxima are produced by scattering elements such as stones, which are randomly buried in the ground, which do not exhibit any prevalent orientation to be revealed by the RT. However, with a simple statistical analysis it is not difficult to discriminate between a Radon transform that exhibits a few highly localized maxima and another that has many low peaks due to random scatterers.

4. Experimental results

The proposed approach has been tested on a variety of synthetic and natural data sets, with voxset sizes ranging from $50 \times 50 \times 70$ to $100 \times 100 \times 200$ voxels.

We first considered synthetic data sets that included two pipes of different diameters, placed in different positions and with different orientations. The diameters in the tests were 3, 5 and 7 voxels, and we used a 5-pixel wide mask for the low-pass filter on the projection planes (see Section 2.3).

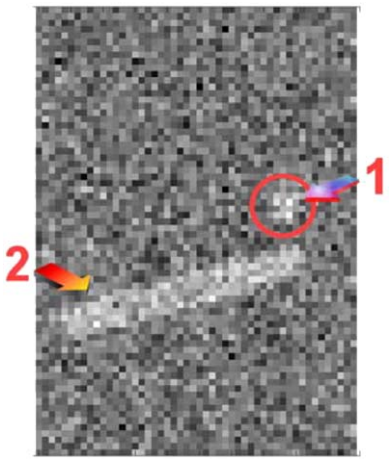

Fig. 12. 2D Vertical section of the synthetic data set used for the tests, that contains two pipes. The pipe 1 is orthogonal to the image plane. Standard deviation of added noise is 50% of the intensity of the pipe's voxels.

We initially set all voxels to zero except for those corresponding to the pipes, which we set to 1. Then we added a Gaussian noise of various intensities, and we measured the errors in the estimation of the position, orientation and diameter of the pipes.

The standard deviation of the superimposed noise was set to 12.5%, 25% and 50% of the voxel values corresponding to the pipes (Fig. 12). The results of this analysis are collected in Table 1, which confirms that the algorithm is capable of good performance even in the presence of a strong noise.

Notice that a better estimation of position and orientation for those pipes that have a diameter of 5 voxels is to be attributed to the size of the filtering mask. The error for pipes of different size, however, keeps rather modest (less than 3° for the angles ϑ and φ, and less than one voxel for the offsets u and v).

As far as the real data sets are concerned, we first tested the effectiveness of both the iterative processing and the termination criterion. We conducted the tests on a 2D version of the algorithm, which emulates the 3D one (Fig. 13) on a horizontal section of a 3D data set. In this experiment, the leading pipe (the horizontal one) is found to be $\theta = 178°$.

The next task consists of identifying the spots that correspond to the pipe. The first considered data set is, in fact, very good, as the entire pipe generates a single elongated spot. After recognizing the side margins of the pipe (see image on the left on the second row of Fig. 13), the target can be removed and a new RT can be computed on the residual. Notice that a pipe tends

Table 1
Pipe detection performance with a synthetic and noisy data set. The angles (θ, φ) characterize the orientation of the 3D line, while (u_0, v_0) characterize the offset of the line from the origin on the XY plane

Noise level (Std Dev.): (% of pipe value)	Diameter (voxel)	Mean error orient: ϑ, φ (deg)	Mean error offset: u, v (voxel)	Mean error diameter (% of real value)
12.5	3	1.12	0.25	2.5
12.5	5	0.38	0.25	3.5
12.5	7	1.37	0.25	5
25	3	1.62	0.5	5
25	5	0.50	0.25	6
25	7	1.12	0.5	10
50	3	1.87	0.75	4.6
50	5	0.62	0.25	10
50	7	2.12	0.75	25

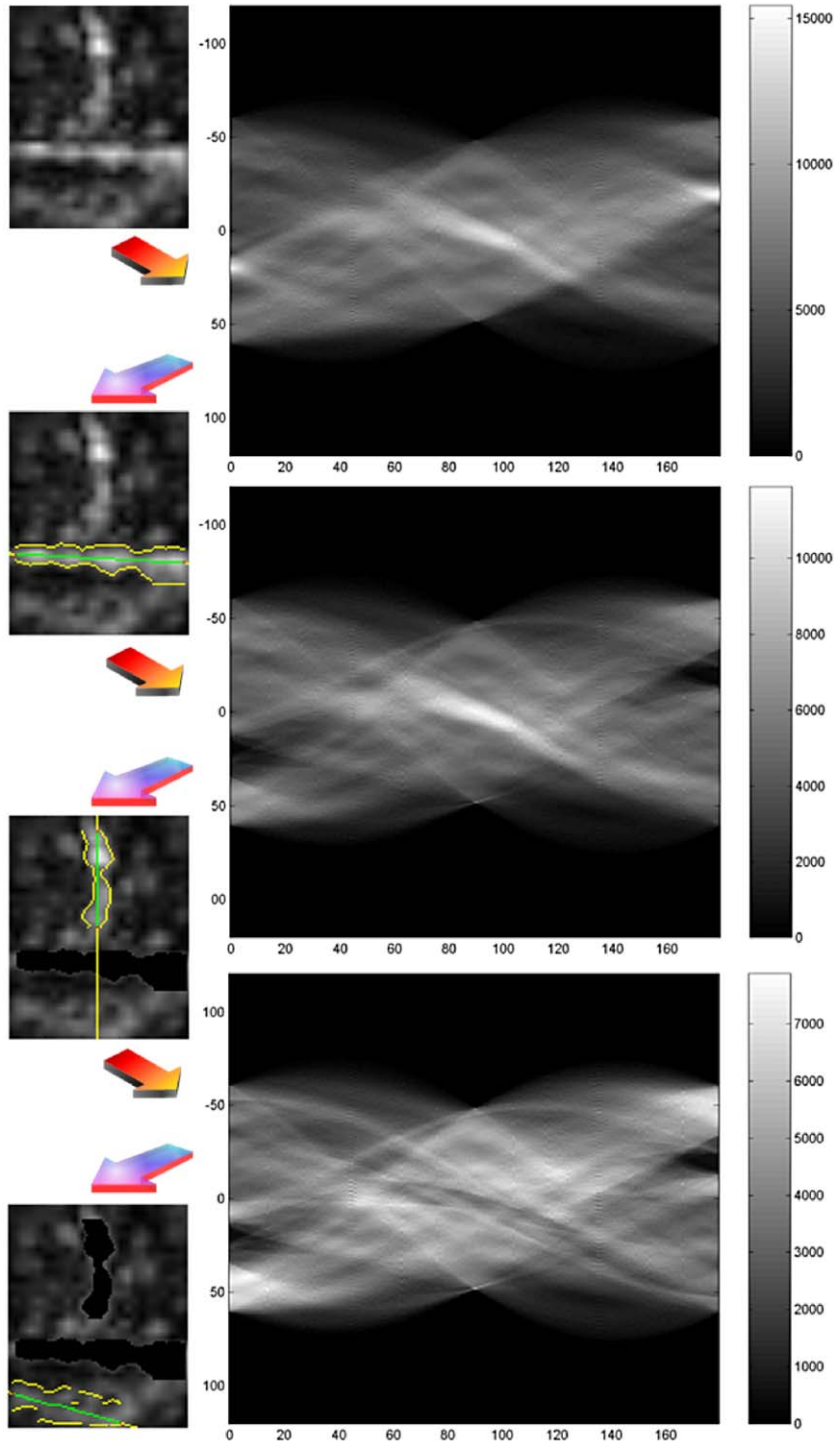

Fig. 13. Three iterations of our algorithm: in the left column the images of the data set before starting the process and after every iteration; in the right column the three (normalized) RTs. Once located the leading target with RT and identified its spots and bounds, the target is cancelled from the data. (NB: due to cyclicity of the θ coordinate, the ρ-axis is re-folded with inverted sign).

to affect the whole plane of the RT, as there are many lines that pass through the blob of the pipe, therefore the removal of data pertaining to just that pipe will affect the whole RT, instead of just the area of the corresponding maximum.

The second iteration is expected to locate the vertical pipe (see left image on the third row of Fig. 13). The target is, once again, made of a single spot that covers only a portion of the line of maximum consensus. After detecting the side margins of the spot, we can remove it from the data and compute a new RT on the residual data. The absolute maximum of the residual data set is found to lie around $\theta \simeq 10°$. This region corresponds to a spurious target, whose margins are marked in the bottom left image of Fig. 13. As the "density" of this target is much lower than that of the two previously detected ones, the algorithm decides to stop, and discards the last detected target (see Section 3.3).

We then tested how the proposed method performs with data sets that tend to originate spurious targets due to badly matched spots. The data set that we used for this experiment (named "Gaggiano") is shown in Figs. 1–3. After computing the RT, the algorithm searches for local peaks and, for each one of them, it computes the coefficients L and H, which are defined in Section 3.2 as Eqs. (2) and (3), respectively. This allows us to discard some of the detected targets.

The intensity profile of the remaining ones is shown in Fig. 10, as well as the values of the coefficients and the parameters of the target line. The first target, which is practically horizontal in the data set, corresponds to a real pipe (pipe 1 in Fig. 3); the second one has a 10° slope with respect to the ground plane, and does not correspond to a real pipe, as confirmed by its projection on the ZY plane shown in Fig. 9 on the left-hand side.

Like the 2D case of Fig. 9 (right-hand side), the 3D RT also shows a maximum for a line of projection that crosses two spots belonging to different pipes. However, the two spots of the spurious pipes are not aligned with the corresponding target line, while the spots of the first one extend along the direction of their target line. In fact, the value of the A coefficient is 0.91 for the correct pipe and 0.01 for the spurious one, and the algorithm can safely choose the target to be extracted and removed. The target lines and the principal axes of the spots are shown in Figs. 14 (real pipe) and 15 (spurious pipe) in solid and dashed lines, respectively. Notice that the second image is a horizontal section of the data set, while the orientation of the target is of $\varphi = 10°$, therefore, the plotted axis and the target line can only be projections of the real ones.

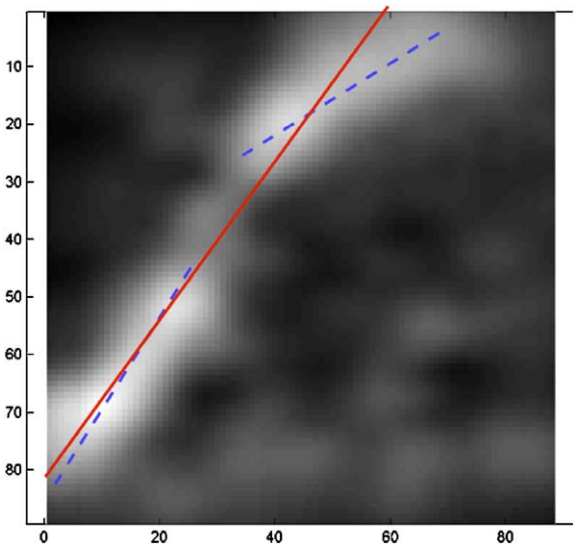

Fig. 14. Target line (solid line) and principal axes (dashed lines) of the spots. Both the spots are well aligned with the target line.

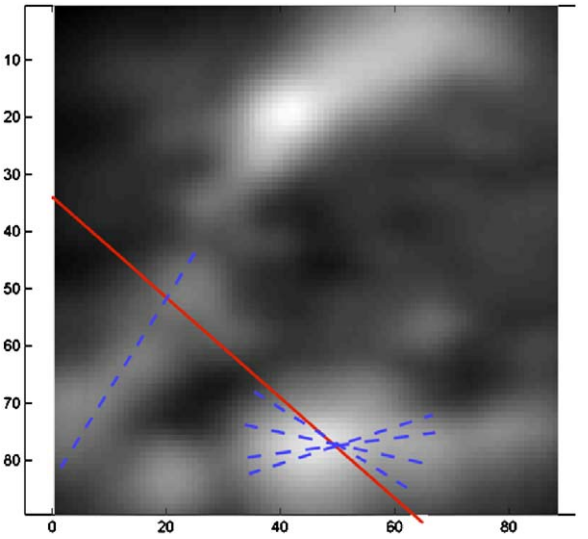

Fig. 15. Target line (solid) and principal axes (dash) of the spots. The left spot has a pronounced principal direction almost orthogonal to the target's line. The bottom right one does not have a principal axis.

The fact that the coefficient A of Eq. (4) involves two non-linear characteristics (shaped like sigmoids) turns out to be very helpful with this data set. In fact, the term σ_1 of Eq. (5), which weighs the scalar product between the target line and the principal axis of each one of the spots, allows misalignments up to 30°. Because of that, the first target is allowed to reach a high value for A, even though the principal axis of one of its spots is pointing 22° away from the direction of the target line because the pipe is slightly bent. The term σ_2, which weighs the alignment's reliability, ignores the information about the alignment of the bottom right spot (see Fig. 15) of the spurious target. In fact, this spot does not have a principal direction, and could turn out to be aligned with the target, with the result of increasing the value of the coefficient A.

This definition of the alignment factor A allows the algorithm to select among all the detected targets the one that is most likely to correspond to a real pipe or beam. After removing this target from the data set, the residual data set becomes simpler to analyze, and many of the spurious pipes detected in the previous iteration may automatically disappear in the next one.

5. Implementation issues

We implemented our algorithm in a Windows environment, using C++ and MATLAB^{TM}. The MATLAB module manages the graphic interface, while a Dynamic Linkage Library (DLL) written in C++ carries out all computational tasks.

In order to optimize the algorithm from the computational standpoint, particular attention has been paid to the implementation of DLL. The result is, in fact, very encouraging, considering the size of the data sets and the number of operations needed to compute the RT. In order to further reduce the computational load we adopted a *multi-resolution* strategy, as suggested for example in [12] in the case of the HT. In order to give an idea of the improvement in the computational efficiency, it suffices to say that a complete scan (with a sampling step of 1° for ϑ and φ, and of 1 voxel for the line offset) of a standard data set takes about 8 min [3] using our optimized algorithm. Doubling the sampling steps, the computational time becomes about 30 s, which is about 16 times less than before.

The multiresolution approach that we propose consists of two steps:

1. scan the data set at a rough resolution and find the local peaks of the RT. Every peak corresponds to a *target* (a possible linear scattering object),
2. choose the most likely target among the available ones, and refine its localization with a "local" higher-resolution RT.

A typical problem of Hough–Radon algorithms is the amount of memory that is needed to compute a complete transform, which grows significantly with the number of parameters that describe the patterns. For example, the memory required to compute just the lower-resolution RT (a data set of $50 \times 50 \times 100$ voxels, with a sampling step of 2° for both ϑ and φ) is about 200 MB. In order to reduce this memory load, we can split the 4D RT into cells of size $10° \times 10° \times 10$ voxel $\times$ 10 voxel, and store in memory only the four parameters of the "maximum consensus" line for each cell (see Fig. 16). In fact, this is the only information that we need to compute the local higher-resolution RT. Using an RT approach, the simplest way to do so is to store the value of the integral along a line only if integration along the previously considered lines of the cell gave us lower values. Conversely, in the HT, the transformed values are obtained by a cumulation process through data scanning, and therefore, the search for the positions of the maxima can be carried out only by looking at the entire transform that was previously computed and stored. For this reason, we based our detection process on a basic RT algorithm, instead of global solutions that work, for example, in the frequency domain RT [19] or the HT.

At the end, the detection of a single target takes about 45 s : 30 s for the computation of the coarse-resolution global RT, 10 s for the computation of the local higher-resolution RT, and 5 for the computation of the coefficients and the selection of the target to extract. Due to the iterative nature of the algorithm, the processing time of a voxset is proportional to the number of targets. Typically, a $100 \times 100 \times 200$ voxset takes a total of less than 5 min to be processed.

[3] CPU equipped with an Athlon AMD processor at 800 MHz and 256 MB RAM.

Fig. 16. Cells of the first half ($0^\circ < \theta < 90^\circ$) of a rough-resolution RT (data set 'Gaggiano'). Every rectangle (6×10 *pixel*) collects the values of the 'maximum consensus' lines in a range of 10° of θ and φ. Rectangles are sorted horizontally for increasing θ and vertically for increasing φ. Every rectangle is formed by 60 cells, and the value of every cell is the value of the 'maximum consensus' line in a range of 10 pixels for the horizontal and vertical offsets (u, v). The cell $50^\circ < \theta < 60^\circ,\ 0^\circ < \varphi < 10^\circ, 4, 4$, corresponds to the pipe 1 of Fig. 3.

6. Conclusions

In this paper we proposed a semi-automatic approach to the detection of linear scattering objects in geo-radar data sets, based on the 3D radon transform (RT).

One of the main features of our algorithm is the robustness and the reliability of the detection of the targets, even in the presence of data affected by heavy noise, artifacts and other undesired scattering objects.

The solution that we propose is iterative, as each detected target is removed from the data set before the next iteration, in order to avoid mutual interference or masking. In addition, the algorithm is able to further analyze the data set in a local fashion in order to eliminate spurious targets from the set of lines of maximum consensus.

The proposed solution proved effective with all the real data sets that we tested it on. We found that some very noisy data sets may give rise to some problems in the identification of spurious targets. In fact, in these cases, real and spurious targets may turn out to produce similar scores when combining the coefficients L, H and A of Eqs. (2), (3) and (4). Even in these rare cases, however, the algorithm will be able to perform well with a minimal human supervision in the selection of the dominant target among a small set of possible ones. This selection can be done, for example, by looking at the magnitude profile along the target lines (see Fig. 10), or by looking directly at the 3D volume along the directions that are suggested by the algorithm, or by basing the choice on some a priori knowledge of the scene.

Future developments of this technique are in the use of a probabilistic framework for the morphological characterization of the spots. For example, all possible sets of spots could be jointly analyzed to identify "valid" associations, corresponding to actual linear scatterers.

References

[1] A.S. Aguado, M.S. Nixon, A new hough transform mapping for ellipse detection, Technical Report, Research Journal 1995/96, Department of Electronics and Computer Science, University of Southamtpon.

[2] W. Al-Nuaimy, H. Lu, S. Shihab, Automatic 3-dimensional mapping of features using GPR, Ninth International Conference on G.P.R. S.Barbara, CA, May 2002.

[3] M. Atiquzzaman, Multiresolution hough transform—an efficient method of detecting patterns in images, IEEE Trans. Pattern Anal. Mach. Intel. 14 (11) (1992) 1090–1095.

[4] D.H. Ballard, Generalizing the hough transform to detect arbitrary shapes, Pattern Recognition 13 (2) (1981) 111–122.

[5] S.R. Deans, Hough Transform from the Radon Transform, IEEE Trans. Pattern and Mach. Intell. 3 (2) (1981) 185–188.

[6] S. Delbò, P. Gamba, D. Roccato, A fuzzy approach to recognize hyperbolic signatures in subsurface radar images, IEEE Trans. Geosci. Remote Sensing, accepted, 2000 (also published as Internal Report n. 29/99).

[7] C. Espinosa, M.A. Perkowski, Hierarchical Hough transform based on pyramidal architecture, Northcon, Portland, 1–3 October 1991, pp. 291–296.

[8] P.F. Fung, W.S. Lee, I. King, Randomized generalized hough transform for 2-D grayscale object detection, Thirteenth International Conference on Pattern Recognition (ICPR), Vol. 3, Vienna, 1996, pp. 511–515.

[9] P. Gamba, S. Lossani, Neural detection of pipe signatures in Ground Penetrating Radar data, IEEE Trans. Geosci. Remote Sensing 38 (2) (March 2000) 790–797.

[10] D.D. Haule, A.S. Malowany, Object recognition using fast adaptative Hough transform, IEEE Pacific Rim. Conference on Comm., Comp. and Signal Proc., June 1–2, 1989, pp. 91–94.

[11] V.E. Henson, M.A. Limber, S.F. McCormick, B.T. Robinson, Spotlight computed tomography with natural pixels, Fifth SIAM Conference on Applied Linear Algebra, 1994, pp. 97–101.

[12] J. Illingworth, J. Kittler, The adaptive hough transform, Trans. Pattern and Mach. Intell. 9 (1987) 690–698.

[13] V. Jean, Fast Linear Hough Transform, International Conference on Application-Specific Array Processors, IEEE Computer Society Press, MD, 1994, pp. 1–9.

[14] A.C. Kak, M. Slaney, Principles of computerized tomographic imaging, IEEE Press, New York, 1988.

[15] R. Krishnapuram, D. Casasent, Determination of three-dimensional object location and orientation from range images, IEEE Trans. Pattern and Mach. Intell. 11 (11) (1989) 1158–1167.

[16] K.M. Lee, W.N. Street, A fast and robust approach for automated segmentation of breast cancer nuclei IASTED International Conference on Computer Graphics and Imaging (CGIM), Palm Springs, CA, USA, October 25–27, 1999, pp. 42–47.

[17] R.A. McLaughlin, Randomized Hough transform: improved ellipse detection with comparison, Pattern Recognition Lett. 19 (3–4) (1998) 299–305.

[18] A. Mukherjee, S.K. Parui, B.B. Chaudhuri, R. Krishnan, K.K. Rao, Detection of linear features in satellite imagery using robust estimation, ICPR (A), Jerusalem, 1994, pp. 514–516.

[19] L. Murhy, Linear feature detection and enhancement in noisy images via the Radon transform, Pattern Recognition Letters 4 (4) (1986) 279–284.

[20] W.K. Pratt, Digital Image Processing, 3rd Edition, Wiley-Interscience, New York, 2001.

[21] A. Sarti, S. Tubaro, Detection and characterization of planar fractures using a 3d hough transform, EURASIP Signal Processing 82 (9) (2002) 1269–1282.

[22] F. Scheck, Mechanics, Springer, Berlin, 1990.

[23] R.E. Sheriff, Encyclopedic dictionary of exploration geophysics, Society of Exploration Geophysicists, 1991.

[24] V.A. Shapiro, On the Hough transform of multilevel pictures, Pattern Recognition 29 (4) (1996) 589–602.

[25] P. Sobrevilla, E. Montseny, J. Keller, Road and Pipe Detection in LADAR Intensity Images through Fuzzy Techniques, IFSA/NAFIPS, Vancouver, Canada, July 2001.

[26] P. Toft, The radon transform—theory and implementation, Ph.D. Thesis, Deparment of Mathematical Modelling, Section for DSP, Technical University of Denmark, June 1996.

[27] H.S. Youn, C.C. Chen, Automatic GPR target detection and clutter reduction using neural network, Ninth International Conference on G.P.R., S. Barbara, CA, May 2002.