Pattern Anal Applic (2018) 21:671–684
https://doi.org/10.1007/s10044-016-0592-5

THEORETICAL ADVANCES

Statistical moments calculated via integral images in application to landmine detection from Ground Penetrating Radar 3D scans

Przemysław Klęsk[1] · Mariusz Kapruziak[1] · Bogdan Olech[1]

Received: 13 April 2015 / Accepted: 20 December 2016 / Published online: 7 January 2017

Abstract Under study is an application of Ground Penetrating Radar (GPR) to landmine detection problem. We focus on the detection of antitank mines carried out in the 3D GPR images, so-called C-scans, by means of a machine learning approach. In that approach, we particularly pursue a technique for fast extraction of image features based on an initial calculation of *multiple integral images*. This allows later to calculate each feature in constant time, regardless of the scanning window position and size. The features we study are *statistical moments* formulated in their 3D variant. We present a comparison of detection results for different sizes and parameterizations of feature sets. All results are obtained from a prototype GPR system of our original construction in terms of both hardware and software.

Keywords Ground Penetrating Radar · Landmine detection · Fast features extraction · 3D statistical moments · Multiple integral images

1 Introduction

In the last decade GPR technology has emerged as a popular research topic. Fields, where GPR applications are being considered or already successfully present, are quite miscellaneous: construction industry, archeology, sedimentology, military technology—to mention a few [11, 21, 24].

It is worth explaining that there are three main types of GPR images (radargrams). The simplest variant is an A-scan, being a single GPR profile defined over the time axis only (directed inwards the ground). A linear collection of A-scans along some direction forms a B-scan. A collection of A-scans over a certain area, which also can be treated as a linear collection of B-scans, forms a C-scan, i.e., a three-dimensional image, with coordinates system typically defined as *across track* × *along track* × *time*. The time axis can be intuitively associated with the depth.

In general, any buried objects, which are non-transparent to GPR waves, produce in C-scans the patterns being combinations of *hyperboloids* (resembling bowl-like shapes). For metal landmines at least two strong hyperboloids are usually visible, related to top and bottom surfaces of the mine casing. On the other hand, plastic mines are typically less clearly visible in the image. They produce thinner and more subtle shapes in radargrams. Sometimes, more details of a mine and its casing can be seen (rendered as smaller hyperboloids), but this depends on several aspects like: antenna system, GPR bandwidth, C-scan resolution, soil type and humidity, mine sizes and ground clutter. Figure 1 presents two examples of C-scans generated by our GPR system and detections of antitank (AT) landmines in them.

As regards algorithmic approaches to mine detection task met in the literature, one should look at them keeping

This work has been partially financed by the Polish Ministry of Science and Higher Education. Agreement no. 0091/R/TOO/2010/12 for R&D Project no. 0 R00 0091 12, dated on 30.11.2010, signed with the consortium of Military Institute of Armament Technology in Zielonka, Poland, and Autocomp Management Sp. z o.o. in Szczecin, Poland.

✉ Przemysław Klęsk
pklesk@wi.zut.edu.pl

Mariusz Kapruziak
mkapruziak@wi.zut.edu.pl

Bogdan Olech
bolech@wi.zut.edu.pl

[1] Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Żołnierska 49, 71-210 Szczecin, Poland

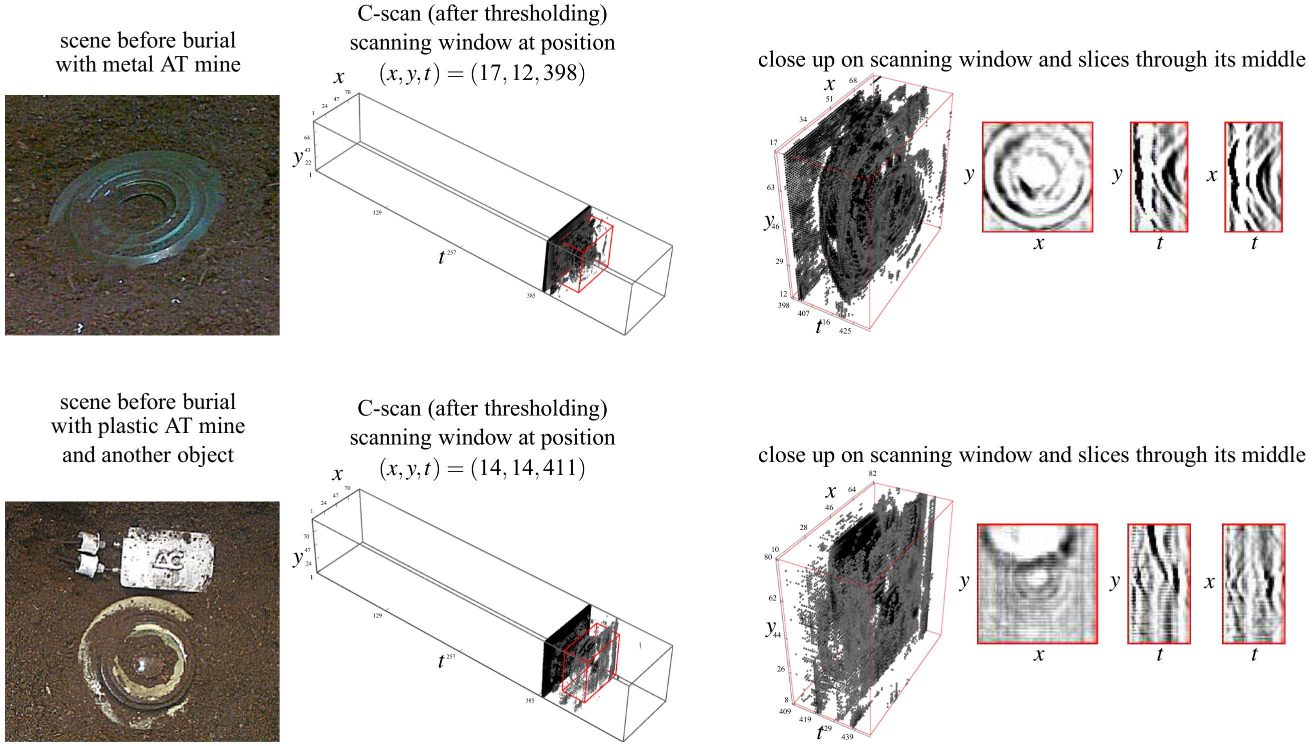
scene before burial
with metal AT mine
C-scan (after thresholding)
scanning window at position
$(x, y, t) = (17, 12, 398)$
close up on scanning window and slices through its middle
x
y
t
scene before burial
with plastic AT mine
and another object
C-scan (after thresholding)
scanning window at position
$(x, y, t) = (14, 14, 411)$
close up on scanning window and slices through its middle

Fig. 1 Two examples of landmine detections in our GPR system: metal AT mine (*top row*), plastic AT mine (*bottom row*). The scene with the plastic mine contains additionally a metal box with cables, serving as a distraction object for the detector

in mind a distinction between two stages: (1) features extraction and (2) learning and classification algorithms. As regards the latter, quite many state-of-the-art methods have been tried out, e.g., Naive Bayes and LVQ in [6], neural networks in [10], least squares curve fitting in [9, 26], HMMs in [14, 17, 26] or ensemble classifiers in [12, 22]. Yet, it seems, in general, that the final success is less dependent on the choice of learning algorithm and more dependent on the quality of images and features extracted from them.

It is worth to mention that the process of features extraction for GPR applications is often accompanied by auxiliary techniques, such as hyperbola or ellipse detection. In order to reduce hyperboloids to hyperbolas or ellipses, the C-scans must be preprocessed and suitable B-scans or time slices must be selected out. For example, Milisavljević et al. [16] detected hyperbolas via Hough transform. Zhu and Collins [26] used polynomial curve fitting. Later, hyperbola characteristics or polynomial coefficients served as features for machine learning. In [26], authors measured also (as additional features) the intensities of diagonal and antidiagonal edges of hyperbolas. As regards the features that can be extracted from ellipses in time slices, Yarovoy et al. [25] measured e.g., horizontal position (from ellipse center), dielectric permittivity of the ground (from the increase in ellipse size), depth of burial (from time delay and calculated ground permittivity).

1.1 Motivation and contribution

The main motivation for our research was to work directly on C-scans and thereby to focus on features describing three-dimensional shapes. Obviously, a dense scanning/detection procedure carried out over a 3D image of high resolution is computationally expensive, because for every position of the scanning window calculations related to features extraction and classification must be performed.

In this paper we make an attempt to apply statistical moments as features. Various applications of 2D statistical moments are known from computer vision—quite many of them met in the field of optical and handwritten character recognition, see e.g., [1, 4], but also in a general object detection setting [7, 13]. We want to check the applicability of 3D statistical moments to landmine detection.

The main contribution of the paper is an idea to speed up the extraction of moments for each image window by means of *multiple integral images*, calculated once, prior to the detection procedure. One may come across publications where a similar idea is applied in 2D cases, especially in the context of variance or covariance calculations [19, 20]. Yet for some reasons, statistical moments of still higher orders, supported by integral images, can hardly be met, although the technique can be extended in a straightforward manner. We derive suitable formulas in the paper.

1.2 Organization of this paper

The rest of this paper is organized as follows. Section 2 pertains to computational aspects of statistical moments in the context of detection tasks. In Sect. 2.1 we shortly review the so-called central statistical moments and define their 3D variant suitable for our images. Section 2.2 demonstrates the main contribution of the paper, namely, the technique to extract the moments fast—in constant time. Sections 2.3 and 2.4 discuss some technical details related to the contribution, such as: preparation of integral images and generation of features by windows partitioning, respectively. Section 3 is the experimental section. It describes an application of the proposed method to landmine detection based on GPR, in particular: the hardware of our prototype radar, measurements collected from different scene variations, feature spaces and data sets, and the machine learning setup. Finally, the section discusses results of tests (10-fold cross-validation) with the focus on: error rates, ROC[1] curves and time performance. Section 4 summarizes the paper.

Additionally, we encourage the reader to study Appendix 1, in which we compare our results against the ones obtained on the same learning material by a benchmark method due to Torrione et al. [22].

2 Statistical moments and integral images

2.1 3D statistical moments

A good intuition on statistical moments (working as image features in recognition or detection tasks) can be gained by thinking first of moments for continuous probability distributions. For the 2D case the central continuous moments weighted by a density function f are

$$\mu^{p,q} = \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} \left(x-\mu^{1,0}\right)^p \left(y-\mu^{0,1}\right)^q f(x,y)\,dx\,dy, \tag{1}$$

where p, q define the moment order variable-wise, and here $p+q \geq 2$; the moments of order one are

$$\mu^{1,0} = \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} x f(x,y)\,dx\,dy, \tag{2}$$

$$\mu^{0,1} = \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} y f(x,y)\,dx\,dy. \tag{3}$$

As regards moments for images, the integrals are replaced by sums weighted by pixel intensities (instead of a density).

In the setting of our landmine problem, we firstly need to account for the 3D case we have, and secondly we need to define moments for image windows (cuboids) not the whole images. Thus, we shall define *3D normalized central moments* independent of the window position and size.

Let i denote the 3D image function (a C-scan). The point value $i(x,y,t)$ represents the image intensity over coordinates (x,y) for the time moment t. For a window spanning from (x_1,y_1,t_1) to (x_2,y_2,t_2) we define the moments of interest as follows

$$\mu^{p,q,r}_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}} = \sum_{x_1\leq x\leq x_2}\sum_{y_1\leq y\leq y_2}\sum_{t_1\leq t\leq t_2} \left(\frac{x-x_1}{x_2-x_1} - \mu^{1,0,0}_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}\right)^p \left(\frac{y-y_1}{y_2-y_1} - \mu^{0,1,0}_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}\right)^q \left(\frac{t-t_1}{t_2-t_1} - \mu^{0,0,1}_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}\right)^r \cdot i(x,y,t)/S, \tag{4}$$

where $S = \sum_{x_1\leq x\leq x_2}\sum_{y_1\leq y\leq y_2}\sum_{t_1\leq t\leq t_2} i(x,y,t)$, and the moments of order one are

$$\mu^{1,0,0} = \sum_{x_1\leq x\leq x_2}\sum_{y_1\leq y\leq y_2}\sum_{t_1\leq t\leq t_2} \frac{x-x_1}{x_2-x_1}\cdot\frac{i(x,y,t)}{S}, \tag{5}$$

$$\mu^{0,1,0} = \sum_{x_1\leq x\leq x_2}\sum_{y_1\leq y\leq y_2}\sum_{t_1\leq t\leq t_2} \frac{y-y_1}{y_2-y_1}\cdot\frac{i(x,y,t)}{S}, \tag{6}$$

$$\mu^{0,0,1} = \sum_{x_1\leq x\leq x_2}\sum_{y_1\leq y\leq y_2}\sum_{t_1\leq t\leq t_2} \frac{t-t_1}{t_2-t_1}\cdot\frac{i(x,y,t)}{S}. \tag{7}$$

We remark that the aforementioned normalization is related to the presence of terms $(x-x_1)/(x_2-x_1)$ (similarly for y, t), due to which our moments take values in the $[-1,1]$ interval.

2.2 Calculations of moments via integral images

Let us now reformulate the moments in terms of integral images and their growths. First, we define a *general 3D integral image* $ii^{p,q,r}$ (of order $p+q+r$) as

$$ii^{p,q,r}(x,y,t) = \sum_{1\leq j\leq x}\sum_{1\leq k\leq y}\sum_{1\leq l\leq t} x^p y^q t^r i(x,y,t). \tag{8}$$

We give an induction algorithm to calculate $ii^{p,q,r}$ in the next subsection.

Next, it is useful to define the *growth* operation for integral images. Growths shall later serve as an economic way (constant time) to calculate sums of suitable moment-related terms weighted by pixel intensities in image windows. In the 3D case, growths can be expressed using only 8 elements of the integral image. For a window spanning from (x_1,y_1,t_1) to (x_2,y_2,t_2) the growth can be defined e.g., as

[1] Receiver Operating Characteristics.

$$
\begin{aligned}
\Delta_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}(ii) = & \; ii(x_2,y_2,t_2) - ii(x_1-1,y_2,t_2)\\
& - ii(x_2,y_1-1,t_2) + ii(x_1-1,y_1-1,t_2)\\
& - \Big(ii(x_2,y_2,t_1-1) - ii(x_1-1,y_2,t_1-1)\\
& - ii(x_2,y_1-1,t_1-1) + ii(x_1-1,y_1-1,t_1-1)\Big),
\end{aligned}
\tag{9}
$$

where ii stands for some integral image.

The following proposition constitutes the main algorithmic contribution of the paper.

Proposition 1 *Given a maximum order $N \geq 0$ of moments, suppose the set of integral images $\{ii^{p,q,r}\}$, $0 \leq p,q,r \leq N$, defined as in (8), has been calculated prior to the detection procedure. Then, for any cuboid in the image, spanning from $(x_1,y_1,t_1)$to (x_2,y_2,t_2), each of its statistical moments can be extracted in constant time—$O(1)$—regardless of the number of pixels within the cuboid, as follows:*

$$
\begin{aligned}
\mu^{p,q,r}_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}} = & \frac{1}{\Delta_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}(ii^{0,0,0})(x_2-x_1)^p(y_2-y_1)^q(t_2-t_1)^r}\\
& \cdot \sum_{j=0}^{p}\sum_{k=0}^{q}\sum_{l=0}^{r}(-1)^{p+q+r-j-k-l}\binom{p}{j}\binom{q}{k}\binom{r}{l}\\
& \cdot \left(x_1 + \mu^{1,0,0}_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}(x_2-x_1)\right)^{p-j}\\
& \cdot \left(y_1 + \mu^{0,1,0}_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}(y_2-y_1)\right)^{q-k}\\
& \cdot \left(t_1 + \mu^{0,0,1}_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}(t_2-t_1)\right)^{r-l}\\
& \cdot \underbrace{\sum_{x_1\leq x\leq x_2}\sum_{y_1\leq y\leq y_2}\sum_{t_1\leq t\leq t_2} x^j y^k t^l i(x,y,t)}_{\Delta_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}(ii^{j,k,l})}.
\end{aligned}
\tag{10}
$$

Proof The proof is in fact a straightforward derivation from formula (4). First, the means (moments of order one), that are present under powers, should be multiplied by suitable unity terms: $\mu^{1,0,0}_{\cdot}\cdot\frac{x_2-x_1}{x_2-x_1}$, $\mu^{0,1,0}_{\cdot}\cdot\frac{y_2-y_1}{y_2-y_1}$, $\mu^{0,0,1}_{\cdot}\cdot\frac{t_2-t_1}{t_2-t_1}$. This allows to extract the denominators and form the normalizing constant $1/((x_2-x_1)^p(y_2-y_1)^q(t_2-t_1)^r)$ in front of the summation. Then, the powers are expanded by means of the binomial theorem, grouping the terms into the ones dependent on the current pixel index (x, y, t), namely the terms: $x^j y^k t^l$, and the ones independent of it. Finally, by changing the order of summations one arrives at the equivalent formula (10). The underbrace indicates how the expensive summation over all pixels in the cuboid is replaced by the constant-time computation (cheap) of the growth of a suitable integral image: $\Delta_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}(ii^{j,k,l})$. Note also that the required normalizer S is calculated by the growth of the zero-order integral image $S = \Delta_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}(ii^{0,0,0})$. □

For the sake of strictness, we should remark that though the calculations involved in (10) are constant time with respect to the number of pixels in a cuboid, they are polynomial with respect to the given moment order, represented by p, q, r. More precisely, the total number of operations is proportional to $(p+1)(q+1)(r+1)$ times seven additions (or subtractions) involved in the growth operator $\Delta_{\substack{x_1,y_1,t_1\\x_2,y_2,t_2}}(ii^{j,k,l})$ as defined in (9).

2.3 Derivation of integral images: induction

The algorithm 1, presented below, is a form of induction and calculates any wanted 3D integral image $ii^{p,r,q}$ from (8) in a single image pass; i.e., in $O(n_x n_y n_t)$ time, where $n_x \times n_y \times n_t$ represents the resolution of a C-scan. Therefore, if one imposes for the moments a maximal order N variable-wise, i.e., $0 \leq p,q,r \leq N$, then there are $(N+1)^3$ integral images to be calculated, and the overall cost becomes $O\Big((N+1)^3 n_x n_y n_t\Big)$.

Algorithm 1 Integral image induction.

```
create three arrays: $ii^{p,q,r}_{n_x\times n_y\times n_t}$, $jj_{n_y\times n_t}$, $kk_{n_t}$.
for $x = 1,\ldots,n_x$ do
    for $y = 1,\ldots,n_y$ do
        for $t = 1,\ldots,n_t$ do
            $a := i(x,y,t)x^p y^q t^r$
            if $t > 1$ then
                $s := kk(t-1) + a$
            else
                $s := a$
            end if
            $kk(t) := s$
            if $y > 1$ then
                $s := s + jj(y-1,t)$
            end if
            $jj(y,t) := s$
            if $x > 1$ then
                $s := s + ii^{p,q,r}(x-1,y,t)$
            end if
            $ii^{p,q,r}(x,y,t) := s$
        end for
    end for
end for
return $ii^{p,q,r}$
```

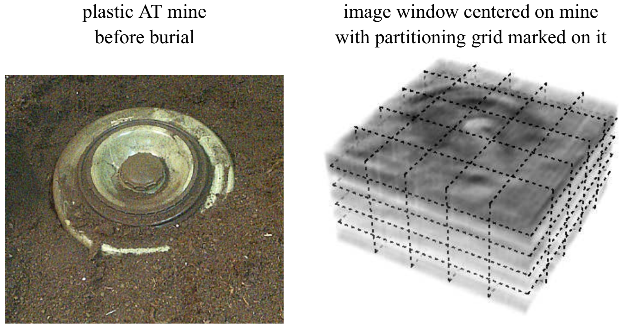
plastic AT mine
before burial
image window centered on mine
with partitioning grid marked on it

Fig. 2 Illustration of the partitioning grid for $m = 5$

2.4 Introducing more features by partitioning image windows

Up to now, we have formulated (for simplicity) the moments as being extracted from whole 3D windows. Given N as the maximal order, this approach implies that the total number of features is $(N+1)^3$. Unfortunately, that is also the number of integral images to be calculated, which for a larger N may constitute a considerable time cost. Recall that the calculation (10) of a single moment, though independent of the number of pixels, scales with p, q, r values. On the other hand, in practice we would like to have many features for learning and the final description of objects, e.g., of order 10^3 or 10^4—as it is common in computer vision applications (for example in face detectors). To resolve this problem we propose a simple operation of window *partitioning*.

Imagine a 3D window is partitioned into a regular $m \times m \times m$ grid of cuboids (later on in our GPR experiments, we try out $m = 3$ and $m = 5$). The moments from now on shall be extracted from each cuboid. This will allow us to have a greater number of features, namely:

$$n = m^3 (N+1)^3, \tag{11}$$

while keeping N (and implied extraction costs) fairly small. An illustration of the partitioning operation is shown in Fig. 2. Looking back at formula (10), one should understand that from now on that, with the partitioning applied, the coordinates x_1, y_1, t_1 and x_2, y_2, t_2 represent suitable bounding coordinates for a single cuboid within the grid (not for the whole 3D window).

3 Measurements, experiments and results

3.1 GPR system and laboratory test stand

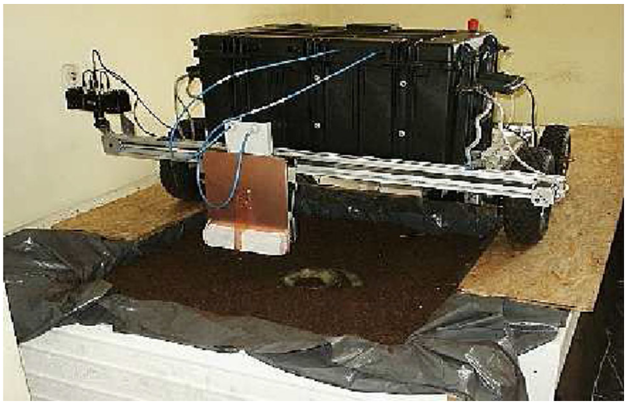

Fig. 3 Mobile GPR platform on in-door laboratory test stand

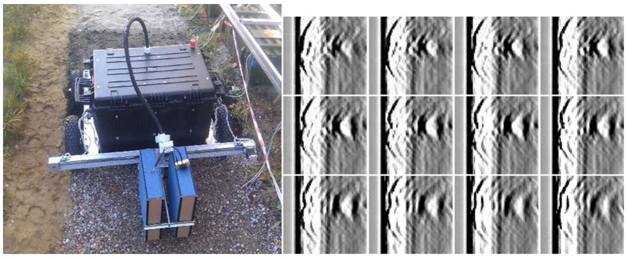

Fig. 4 Out-door test lane and exemplary B-scans of a mine along track collected over gravel

In our research project we have constructed a mobile GPR platform shown in Figs. 3 and 4. The platform contains the antenna system and a standard VNA (Agilent E5071C, inside the black case) as the core of the GPR. Successive B-scans are performed by the platform perpendicularly to its movement. The motion of the platform is remotely controlled by a joystick. Raw data from the scanning are transferred to a host computer through WiFi. The host is a standard PC with a server configuration (Xeon 2.4 GHz 2 × 8-core, 64-bit 24 GB RAM, 2 TB of disk space), also equipped with an nVidia Tesla Quadro 6000 for extra computing power and graphics acceleration.

Stepped frequency continuous wave modulation was performed using sequentially generated commands of S-Parameters measurement transmitted to the VNA for any next frequency. Typically, for SFCW radars [18], the amplitude/phase responses are gathered for each discrete frequency transmitted. An appropriate number of these frequencies, covering an effective bandwidth, is needed to achieve required resolution for an A-scan. In our case the effective bandwidth was 12.7 GHz and was limited by the antenna system.

We use our own original antenna system. The transmitting antenna is a form of the Vivaldi type [2], and the receiving antenna has the shielded loop form [3]. The Vivaldi antenna gives good efficiency and directivity having a big enough aperture to cover a sufficient area with homogeneous lighting of microwaves. The loop antenna acts as a point field sensor with small internal ringing.

All the software for control, communication, learning and detection has been implemented by us in the C# programming language.

3.2 Measurements and scene variations

For convenience reasons, the main series of measurements meant to constitute the learning material was performed in in-door conditions over a container (of area $\approx 1\,\mathrm{m}^2$) filled with a garden type of soil. Nevertheless, we should remark that our GPR vehicle has also been tested in out-door conditions, performing scans over four types of soil: peat, garden, sand, gravel. In all cases we managed to obtain suitably clear images, see an example in Fig. 4.

The objects of interest were two AT landmines: a metal one (TM-62M, height 128 mm, diameter 320 mm) and a plastic one (PT-Mi-Ba III, height 110 mm, diameter 330 mm). In the measurements we have also included negative objects, such as: metal cans and boxes, a large metal box with cables, a large round metal disk, a long metal shaft, a wooden box and building bricks. They were meant as disruptions and potential sources of mistakes for the detector. Examples of scanned scenes are depicted in Fig. 5.

The elevation of the antenna over the ground varied from 10 to 15 cm. As regards mine placements in the scenes, we varied their depths of burial from ≈ 0 cm (flash buried) up to 15 cm and their inclination angles approximately in the range $0° \pm 45°$ in different directions. Mines lying flat or almost flat ($0° \pm 15°$) were however the most frequent in the collected material (as it is their natural way of placement).

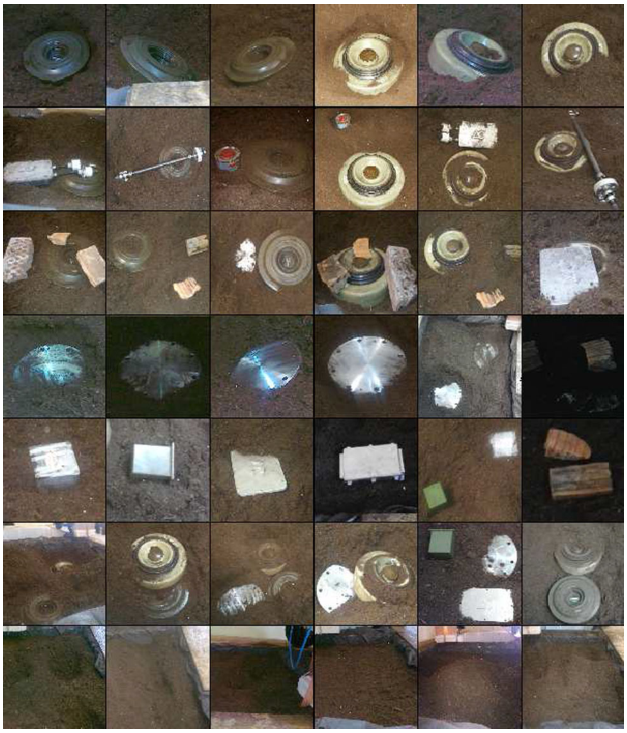

Fig. 5 Examples of scanned scenes before burial and variations on soil surface (*bottom row*)

We should mention that additionally we have experimented with different variations related to the surface of the ground after the objects were buried. Most of the scans were taken with the surface naturally shaped, but we have included also two other extremes: some scenes with the surface flattened down unnaturally with a shovel, and some scenes with unnaturally uneven surface with multiple holes, knolls or canyon-like shapes. Some of these variations are shown in the most bottom row of Fig. 5. It is known in GPR studies that strong surface variations may cause significant changes in the image (especially for high resolutions), propagating onto deeper time slices. Some of such image changes might even be mistaken for an actual object in an extreme case.

3.3 Experimental setup, data sets, learning algorithm

As the learning material collected was a set of 210 C-scans with a physical resolution of 1 cm (distance between two closest A-scans) and image resolution of $92 \times 91 \times 512$ (area of about $1\,\mathrm{m}^2$). The whole material (210 scans) consisted of three groups: 70 scans with the metal mine (and possibly other objects), 70 scans with the plastic mine (and possibly other objects), 70 scans with non-mine objects only.

After some preliminary experimentations, we have decided to thoroughly test four sets of features (3D statistical moments), implied by the following parameterizations:

A. $N = 2$, $m = 3$ (total no. of features: $n = 729$),
B. $N = 3$, $m = 3$ (total no. of features: $n = 1728$),
C. $N = 2$, $m = 5$ (total no. of features: $n = 3375$),
D. $N = 3$, $m = 5$ (total no. of features: $n = 8000$).

We shall use the A, B, C, D naming of the feature sets when reporting the results.

A 10-fold cross-validation scheme was introduced. In every fold a testing pack consisted of: 7 metal mine scans, 7 plastic mine scans and 7 non-mine scans. Training packs were suitably 9 times larger, each containing 189 scans. Before the actual learning, each training pack was processed in a batch manner (images traversed with a scanning 3D window) and transformed to a data set consisting of multiple examples of positive and negative windows. The scanning window was of dimensions: $w_x = 67 \times w_y = 67 \times w_t = 39$, and the traversal procedure was of full density, i.e., with one pixel shifts for the window $dx = dy = dt = 1$. Additionally, the window was allowed to move partially outside the image for x, y variables, so that hyperbolic patterns of mines located near image borders could be sampled more appropriately (more centrally). Such overlaps onto the margins were set up to be equal at most 15% of the scanning window widths. Positive windows were memorized on the basis of positive object coordinates in the images, which we kept registered in an auxiliary file. Beforehand, the process of marking (determining) these coordinates was done visually by a human after each C-scan was taken (supervised learning). We have introduced a 2-pixel tolerance around a target for each x, y, t variable when memorizing positive windows. For negative window examples we had to use undersampling due to their great number. Please note that the majority of negative windows are repeated examples of the ground background, similar among each other; therefore, there is no need to memorize all of them. Such proceedings resulted in generation of large training sets with approximately 10 000 positive and 90 000 negative window examples for each cross-validation fold.[2]

The large number of learning examples and large feature spaces (up to 8000 features) made our machine learning settings similar to the ones known for example from training of face or body detectors. Therefore, we limited our selection of a learning algorithm only to boosting methods with several simple weak learners. It is known that boosting is well suited for large-scale data. Its properties like stagewise progression and mathematical connections to logistic regression make boosting strongly resistant to overfitting [8].

After initial experimentations with such algorithms as: AdaBoost + decision stumps, AdaBoost + bins, RealBoost + normal approximations, RealBoost + bins, RealBoost + decision trees, we have finally decided for that last variant. Observed error rates and ROC characteristics indicated that the RealBoost with decision trees was suited best for the characteristics of our GPR data. We have implemented shallow trees with at most 4 terminal nodes, trained by means of the well-known Gini index as the impurity criterion [8, 15, 23]. The final ensembles (for each CV fold) consisted of 600 weak classifiers.[3] This potentially made the ensemble use at most 1800 features, since each 4-terminal tree involves three inequality tests. In practice we have observed that about 1500 distinct features were present in an ensemble after the learning was finished.

To speed up the boosting procedure itself, we have also implemented the *weight trimming* technique described in [8]. After this modification the learning times for the most numerous features set D (8000 features) were ≈ 2.0 h per fold, as opposed to ≈ 35 h without weight trimming.

3.4 Results

For each set of features we have trained two separate detectors, one aimed to detect only the metal mine—let us refer to it in short as the **“metal detector”**—and the second aimed to detect only the plastic mine—the **“plastic detector”**.[4] At the testing stage, both detectors have been run separately on every test image.

Detailed results of cross-validation are gathered in Table 1. In the table, cells reporting percentages of correct positive detections are marked with a “sensitivity” label, while cells reporting percentages of incorrect positive detections are marked with a “FAR” label (false alarm rate). We distinguish two types of false alarms: (1) proper false alarms—e.g., when a plastic mine or a non-mine object is detected incorrectly as a metal mine (or vice versa), (2) side false alarms—e.g., when some window in the image gets detected as a positive one, however it is not correctly focused on a positive object but rather on its side traces or deeper time slices (“echos” of a mine). The second type can be regarded as non-dangerous false alarms, since they accompany the actual correct detections.

In Fig. 6 we show a comparison of ROC curves for different sets of features, averaged over all CV folds. The curves are calculated at the windows level of detail, i.e., each window example is treated as a separate object under detection. In the plot, ranges for both sensitivity and FAR axes were purposely narrowed down to show better the differences between the curves. In the plot legend we also report the AUC[5] measures. $\text{AUC}_{0.1}$ represents the normalized area obtained up to the FAR of 0.1—this indicates how fast the curve grows in its initial stage (a property important in detection tasks[6]). The AUC notation

[2] A 2-pixel tolerance sidewise (along each coordinate) was introduced for the scanning window with respect to a positive target. That is, a single landmine target was typically represented in the training set by a cluster of 125 slightly shifted windows.

[3] The finally selected number of weak classifiers (600) was evolved experimentally, based on observing ROC curves and their AUC measures. The tempo of improvement in AUC measures was negligible after that point, and it made little sense to add more weak classifiers.

[4] Such an approach was dictated by substantially different responses (traces) produced in radagrams by metal and plastic AT mines, see example in Fig. 1. By analogy, a combined single “face or hand” detector would be prone to perform worse than individual “face-only” or “hand-only” detectors.

[5] Area under the ROC curve.

[6] The operating decision threshold of any detector is typically set up to be high (resides in the initial part of an ROC) in order to reduce the number of windows switched on falsely. Obviously, this reduces also the sensitivity, but since a cluster of multiple windows (125 in our case) represents a single mine target, it is sufficient that the procedure detects at least one window in a cluster. When multiple windows lying in close vicinity are detected, they are grouped to be displayed as a single one (a typical postprocessing step in computer vision).

Table 1 Cross-validation results for different sets of features

Metal detector			Plastic detector		
Object type	Detected as metal mine	Additional side false alarms	Object type	Detected as plastic mine	Additional side false alarms
Features set A ($n = 729$)					
Metal mine	**67/70** **95.71%** **(Sensitivity)**	1/70 1.43% (FAR)	Metal mine	9 / 70 12.86% (FAR)	3 / 70 4.29% (FAR)
Plastic mine	1 / 70 1.43% (FAR)	0 / 70 0.00% (FAR)	Plastic mine	**60/70** **85.71%** **(Sensitivity)**	0 / 70 0.00% (FAR)
Other	13 / 70 18.57% (FAR)	0 / 70 0.00% (FAR)	Other	5 / 70 7.14% (FAR)	2 / 70 2.86% (FAR)
Total FAR: 15/210 ≈ *7.14%*			**Total FAR: 19/210 ≈ *9.05%***		
Features set B ($n = 1728$)					
Metal mine	**66/70** **94.29%** **(Sensitivity)**	1 / 70 1.43% (FAR)	Metal mine	8 / 70 11.43% (FAR)	2 / 70 2.86% (FAR)
Plastic mine	1 / 70 1.43% (FAR)	0 / 70 0.00% (FAR)	Plastic mine	**63/70** **90.00%** **(Sensitivity)**	0 / 70 0.00% (FAR)
Other	10 / 70 14.29% (FAR)	0 / 70 0.00% (FAR)	Other	7 / 70 10.00% (FAR)	1 / 70 1.43% (FAR)
Total FAR: 12/210 ≈ *5.71%*			**Total FAR: 18/210 ≈ *8.57%***		
Features set C ($n = 3375$)					
Metal mine	**67/70** **95.71%** **(Sensitivity)**	0 / 70 0.00% (FAR)	Metal mine	4 / 70 5.71% (FAR)	0 / 70 0.00% (FAR)
Plastic mine	1 / 70 1.43% (FAR)	0 / 70 0.00% (FAR)	Plastic mine	**64/70** **91.43%** **(Sensitivity)**	0 / 70 0.00% (FAR)
Other	4 / 70 5.71% (FAR)	0 / 70 0.00% (FAR)	Other	6 / 70 8.57% (FAR)	1 / 70 1.43% (FAR)
Total FAR: 5/210 ≈ *2.38%*			**Total FAR: 11/210 ≈ *5.24%***		
Features set D ($n = 8000$)					
Metal mine	**68/70** **97.14%** **(Sensitivity)**	0 / 70 0.00% (FAR)	Metal mine	5 / 70 7.14% (FAR)	0 / 70 0.00% (FAR)
Plastic mine	1 / 70 1.43% (FAR)	0 / 70 0.00% (FAR)	Plastic mine	**65/70** **92.86%** **(Sensitivity)**	0 / 70 0.00% (FAR)
Other	5 / 70 7.14% (FAR)	0 / 70 0.00% (FAR)	Other	7 / 70 10.00% (FAR)	1 / 70 1.43% (FAR)
Total FAR: 6/210 ≈ *2.86%*			**Total FAR: 13/210 ≈ *6.19%***		

without a subindex stands for the area over the whole [0, 1] range of FAR.

To comment on the results the following remarks can be given. Overall, the results of “metal detectors” are noticeably better than of “plastic detectors.” This is intuitive—responses of plastic mines produce more subtle and weaker traces in images. Also as expected, more numerous sets of features C, D performed much better than A, B sets. In the case of “metal detectors,” this difference is visible mainly in FAR values (the sensitivities remained close to equal). For “plastic detectors,” the difference can be seen both in sensitivity and FAR.

For the largest features set D the results were as follows. The **“metal detector”** yielded a **sensitivity** of $\mathbf{68/70 \approx 97.14\%}$ and in total $\mathbf{6/210 \approx \mathit{2.86\%}}$ **false alarms**. The **“plastic detector”** yielded the **sensitivity** of $\mathbf{65/70 \approx \mathit{92.86\%}}$ and $\mathbf{13/210 \approx \mathit{6.19\%}}$ **false alarms**. As regards the reasons for misdetections, they were mainly caused by mines placed with a significant inclination angle (close to 45°). As regards the false alarms, their most frequent sources were: the large metal disk (see the fourth row in Fig. 5)—similar in size and shape to actual mines. Also, a few false alarms occurred due to certain particular arrangements of other objects, generating some resemblance to mines in the image. It is also fair to comment that the material collected by us did not contain “empty square meters” as it is often the case in out-door test lanes. By that we mean square meters with no objects in the scans, just the soil clutter. In our experiments, every scan (every square meter) contained some object buried. Therefore, the calculated false alarm rates should be regarded as overly pessimistic and would be smaller in more realistic conditions with fewer disruptive objects.

Thinking for a moment of the recognition task (rather than detection) the results can be assessed as satisfactory. The “metal detector’ mistook only once (1 / 70) a plastic mine for a metal one, while the “plastic detector” was slightly worse in this aspect—4 / 70, 5 / 70 mistakes, respectively, for C and D sets of features.

3.5 Time performance of dense detection procedure

Table 2 summarizes the time performance of our detectors measured on our CPU (Xeon 2.4 GHz 2 × 8 core). In the table we report times both without and with parallelization.

We should remark that at the detection stage images were scanned less densely (jumps of the window set to $dt = 1,\ dx = dy = 2$) than it was at the data acquisition stage ($dt = dx = dy = 1$). Moreover, we have restricted the analysis only to a subinterval of time slices, for $t = 381, \ldots, 480$, related to potential mine locations (subsurface or flash buried) with some overhead. Therefore, unnecessary slices have been discarded. Despite these reductions, our scanning loop should still be regarded as computationally expensive, close to an exhaustive C-scan traversal. The loop involved an analysis of approximately 34 000 windows (3D) and for each window the extraction of features and the classification calculations (by 600 boosted decision trees) were performed. For the richest features set D, the duration times of the overall detection procedure were on average 13.7 s long per a C-scan. Given the number of windows analyzed ($\approx 3.4 \cdot 10^4$), this yields the **mean time of analysis for a single window** of approximately **0.40 ms**, which in our opinion is a satisfactorily fast result.

The parallelization led to an about 7 times speed up with respect to sequential calculation. The parallelized elements were: calculation of multiple integral images and the main scanning loop (coarse-grained). Yet, it is fair to add that parallelization, as such, plays a minor role in the algorithmic sense, and the crucial element is integral images. We remind they allow for extraction of each statistical moment in constant time, without the need to iterate over all pixels in each 3D window [(formulas (9), (10)].

It is possible to give an estimate on time performance if the calculations were to be carried out *without* integral images. In that case the computational complexity is

$$O\Big((n_x - w_x + 1)(n_y - w_y + 1)(n_t - w_t + 1) \cdot n_f\big(w_x w_y w_t c_{fe/p} + c_{d/f}\big)\Big), \tag{12}$$

where $(n_x - w_x + 1)(n_y - w_y + 1)(n_t - w_t + 1)$ accounts for the maximum number of window positions, n_f is the number of features to be extracted, $c_{fe/p}$ represents the cost of the extraction per single pixel and $c_{d/f}$ represents the cost of detection (classification) procedure per single feature. Now, even with an optimistic setup of $c_{fe/p} = 10^{-9}s$ and $c_{d/f} = 10^{-9}s$ and for $3.2 \cdot 10^4$ windows one may check that the total time of detection procedure for a single C-scan becomes[7] approximately 8 900 s, thus almost 2.5 hours. Even after parallelization this time is inacceptable in practice. Please note that owing to integral images one simply avoids in (12) the term $w_x w_y w_t$ which is proportional to the number of pixels.

4 Summary

We have reported experimental results obtained by our prototype GPR system for automatic landmine detection. In this paper, we have focused more on computational aspects

[7] Window widths set to $w_x = w_y = 67$, $w_t = 39$ and the number of extracted features $n_f = 1500$ as we had on average with 600 4-terminal trees.

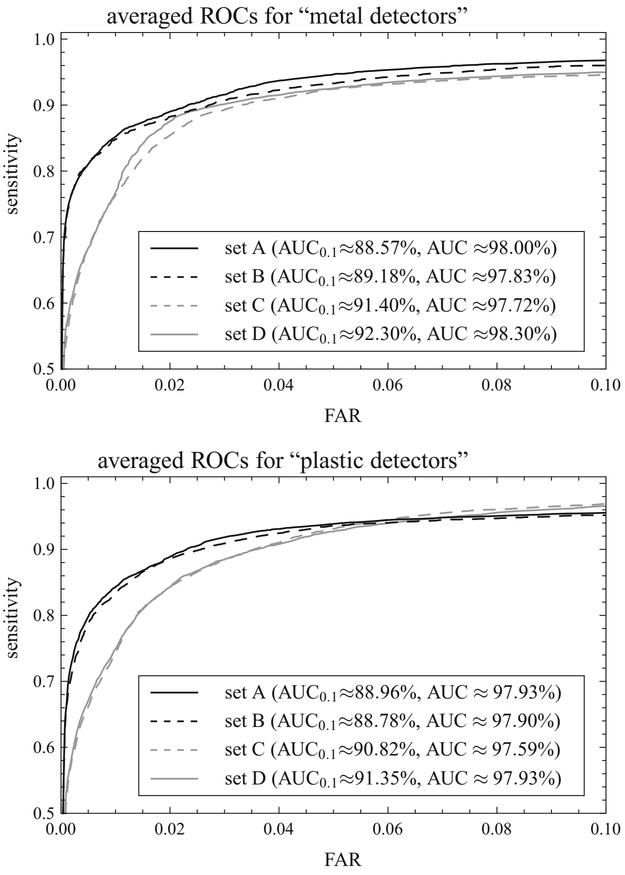

Fig. 6 Comparison of ROC curves (averaged over 10 CV folds) for different sets of features and both types of detectors. The sensitivity and FAR are calculated on the windows level of detail

of the application rather than the hardware ones. A particular attention has been given to fast extraction of features

As the key contribution we regard the technique based on multiple integral images allowing for constant-time calculation of 3D statistical moments. The technique is general and may be applied in computer vision applications (detection tasks) other than ours.

As regards our specific GPR experiments, the technique is helpful in two places. Firstly, at the data acquisition stage, it allows us to generate very large sets of features to learn from. In other words, the learning algorithm is given a rich multitude of features and can look for a relevant subset among them—i.e., such features that describe best the hyperboloids related to mines. Secondly, at the detection stage, we perform a dense traversal of a C-scan (analyzing over 34 000 windows per $\approx 1\,\mathrm{m}^2$) and the constant-time extraction of each statistical moment allows us to carry out the procedure within a reasonable time. Note that we purposely perform no auxiliary operations like: preliminary segmentation, hyperbola detection or prescreening.

The future research direction for us is a more through experimental work. Up to now our results are promising, but it is fair to remark they have been obtained on a fairly small GPR material and with only two types of antitank mines. We have strived to make this material more difficult by introducing many disruptive objects. The future experiments should include more mine types (antipersonnel mines in particular), more soil types and various weather/humidity conditions.

Appendix: Benchmark based on HOG descriptor

In this section we report results obtained on our GPR data by a selected benchmark method. It is a recent method, published in 2014 in IEEE Trans. on Geoscience and Remote Sensing by Torrione et al. [22]. At the features extraction stage the authors of [22] apply the *histogram of oriented gradients* (HOG). As the learning algorithm they apply the *random forest* (RF), originally due to Breiman [5]. Results reported in [22] come from a large US test site of $\approx 200{,}000\,\mathrm{m}^2$ area. The data included 2960 target encounters (mine or other explosives) over 740 unique targets (vehicle was driven four times over the same roads). Reported were: $\approx 95\%$ sensitivity, $\approx 0.0048\mathrm{FA}/\mathrm{m}^2$ FAR, indicating the high effectiveness of the method. We first describe briefly after [12, 22] how HOG features are extracted; then, we report the results obtained on our data. We programmed the benchmark in C# and integrated it with our software.

HOG features

HOG features are 2D features based on gradient angle distributions. The authors of [22] work with C-scans but extract HOG features from B-scans, both across and along track. For brevity, we give formulas only for the across track case ($x \times t$). The $y \times t$ case is analogical.

Let $i(x, t)$ denote a B-scan under analysis. First, the image is convolved with gradient estimation filters: $h_x = (-1, 0, 1)$, $h_t = (-1, 0, 1)^T$. Let $g_x = i * h_x$, $g_t = i * h_t$ represent gradient images. The gradient magnitude at each pixel (j, k) is

$$G(j,k) = \sqrt{{g_x}^2(j,k) + {g_t}^2(j,k)}. \tag{13}$$

Table 2 Time performance for different feature sets (Xeon 2.4 GHz 2 × 8 core)

C-scan resolution with margins: $112 \times 111 \times 512$; window resolution: $67 \times 67 \times 39$; window jumps: $dt = 1$, $dx = dy = 2$ time slices analyzed: $t = 381, \ldots, 480$; no. of windows analyzed: $\approx 3.4 \cdot 10^4$

	Sequential computation	Parallelized computation
Features set A ($n = 729$)		
27 Integral images calculation time	0.74 s	0.16 s
Detection procedure mean time	30.8 s	4.3 s
Detection procedure mean time per window	0.91 ms	0.13 ms
Detection procedure mean time per window per classifier's feature	0.61 μs	0.09 μs
Features set B ($n = 1728$)		
64 Integral images calculation time	1.87 s	0.31 s
Detection procedure mean time	74.6 s	11.1 s
Detection procedure mean time per window	2.19 ms	0.33ms
Detection procedure mean time per window per classifier's feature	1.46 μs	0.20 μs
Features set C ($n = 3375$)		
27 Integral images calculation time	0.74 s	0.18 s
Detection procedure mean time	57.9 s	8.5 s
Detection procedure mean time per window	1.7 ms	0.25 ms
Detection procedure mean time per window per classifier's feature	1.13 μs	0.17 μs
Features set D ($n = 8000$)		
64 Integral images calculation time	1.83 s	0.35 s
Detection procedure mean time	96.1 s	13.7 s
Detection procedure mean time per window	2.83 ms	0.40 ms
Detection procedure mean time per window per classifier's feature	1.89 μs	0.27 μs

Now, for each pixel one should calculate the dominant angle $\theta(j,k)$ of the gradient. In [22], the authors discuss two possibilities for the angle range: $[0,\pi]$ or $[0,2\pi)$. The choice boils down to whether one wants to take the orientation of the gradient vector into account or neglect it. The true gradient points from a darker toward a lighter image region, thus using the full $[0,2\pi)$ angle range. Often though, it is of little importance if an object is darker than the background or vice versa. The authors of [22] choose to follow this simplification and calculate the angle as $\theta(j,k) = \mathrm{atan}(\mathrm{g_t(j,k)/g_x(j,k)})$. This yields the angle within $(-\pi/2, \pi/2)$, by the definition of the atan function, but can be shifted to $(0,\pi)$ for convenience.

For the purpose of comparison, in our experiments we decided to test both possibilities for the angle range. In the case of full $[0,2\pi)$ range, the angle $\theta(j,k)$ is calculated via the atan2 function, commonly available in mathematical libraries.

As remarked in [22], although individual $\theta(j,k)$ and $G(j,\,k)$ can be highly variable (even for similar images), their aggregate statistics over certain image regions (further on referred to as *cells*) provide robust descriptors of those regions. Consider a discretization of angles into n_θ bins of equal width. Imagine that each pixel $(j,\,k)$ votes for the bin its angle $\theta(j,k)$ belongs to, with the magnitude of vote proportional to $G(j,\,k)$. Then, the normalized sums of votes provide the mentioned statistics.

Let the border angles of bins be defined as:

$$\phi_l = l\pi/n_\theta, \tag{14}$$

$$\phi_l = l2\pi/n_\theta - \pi/n_\theta, \tag{15}$$

respectively, for the cases of $[0,\pi]$ and $[0,2\pi)$ ranges. In the second case we make the middle of the first bin (from ϕ_0 to ϕ_1) coincide with the horizontal axis and take into account the radial looping (e.g., the $-\pi/n_\theta$ angle corresponds to $2\pi - \pi/n_\theta$). The vote matrix of dimensionality $n_x \times n_t \times n_\theta$ is:

$$V(j,k,l) = \begin{cases} G(j,k), & \text{if } \phi_{l-1} \le \theta(j,k) < \phi_l; \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Aggregation of votes over a particular cell c is done by summations for each bin index $l = 1, \ldots, n_\theta$:

$$H_1(c,l) = \sum\nolimits_{(j,k)\in c} V(j,k,l). \tag{17}$$

Finally, HOG values for each cell c are derived from H_1 values via normalization taken over the set $N(c)$ of cells being immediate neighbors of c (a.k.a. a *block* of cells):

$$H(c,l) = H_1(c,l) \Big/ \sum\nolimits_{c_q \in N(c)} \sqrt{\|H_1(c_q)\|_2^2 + \epsilon^2}, \qquad (18)$$

where $H_1(c) = (H_1(c,1), \ldots, H_1(c,n_\theta))$. The goal of normalization is to introduce robustness to local ambient changes.

As in [22], we impose a regular 4×3 grid of cells from which to extract the features (4 locations for the spatial axis x or y and 3 for the time axis). Overlapping regions of 3×3 cells constitute blocks for the normalization (18). Since the extraction is repeated for B-scans (crossing the middle of the scanning window) both across and along track, the total number of features is twice the grid size times the number bins: $n = 2 \cdot 4 \cdot 3 \cdot n_\theta$. In other words, the full vector of HOG features is a concatenation of $H(c, l)$ values for all cells, all bins and two B-scan orientations. In tests we imposed: $n_\theta = 9$ for the $[0, \pi]$ angle range (as it is the case in [22]) resulting in $n = 216$ features; $n_\theta = 36$ for the $[0, 2\pi)$ angle range, resulting in $n = 864$ features. We shall refer to these two cases as "variant I" and "variant II," respectively. In Figs. 7 and 8 we visualize HOG features for the two variants. Analogical visualizations can be found in [22].

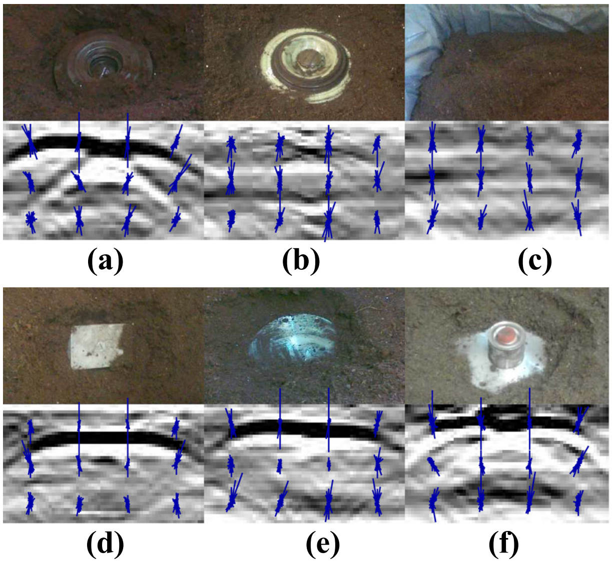

Fig. 8 Sample visualizations of HOG features over B-scans across track with $n_\theta = 36$ bins and angles restricted to $[0, 2\pi)$ (true gradients). **a** Metal AT mine. **b** Plastic AT mine. **c** Soil clutter. **d** Metal box. **e** Metal disk. **f** Looks like mine

Benchmark results and discussion

Training of the detectors (in each CV fold) was carried out by the RF algorithm, with 100 trees as proposed in [22]. The trees in RF are not restricted by maximum depth (contrarily to our approach). Tables 3 and 4 summarize the CV results.

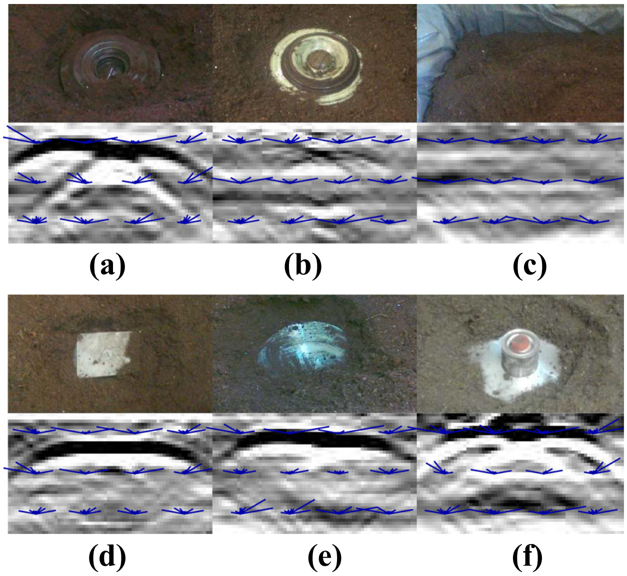

Fig. 7 Sample visualizations of HOG features over B-scans across track with $n_\theta = 9$ angle bins and angles restricted to $[0, \pi]$. **a** Metal AT mine. **b** Plastic AT mine. **c** Soil clutter. **d** Metal box. **e** Metal disk. **f** Looks like mine

Overall, the results indicate an inferior performance of the HOG+RF approach on our GPR data compared to the approach based on 3D statistical moments (3D SM, for brevity). To fix the attention we shall compare HOG+RF against 3D SM with the richest set of features D (Table 1, bottom).

As regards the "metal detector" case, the results were noticeably worse: 66/70 sensitivity for HOG+RF variant II (vs 68/70 for 3D SM) with 19/210 as the total FAR (vs 6/210 for 3D SM). As regards the "plastic detector" case the results were clearly worse: 55/70 sensitivity for HOG+RF variant II (vs 65 / 70) with 26/210 as the total FAR (vs 13/210). In particular, the HOG+RF "plastic detector" turned out to be susceptible to numerous side false alarms.

One should *not* conclude from the results that the HOG-based approach is in itself inferior to the approach based on statistical moments. Rather, it is our GPR data that is demanding due to: numerous disruptive objects, scenes arranged to be difficult (some of with the purpose to generate a response resembling a mine) and lack of empty scenes with no objects, just the soil. Possibly, the HOG-based approach with the setup as was tested uses too few features while learning to cope with these data—$n = 216$ or $n = 864$ as opposed to e.g., $n = 8000$ in our richest setup. It is thinkable that extraction of much more HOG features from a greater number of B-scans (not only 2 middle ones), thus covering more of the 3D window, would

Table 3 10-fold cross-validation results for HOG+RF (variant I)

Metal detector			Plastic detector		
Object type	Detected as metal mine	Additional side false alarms	Object type	Detected as plastic mine	Additional side false alarms
Metal mine	**64/70** **91.43%** **(Sensitivity)**	1/70 1.43% (FAR)	Metal mine	1/70 1.43% (FAR)	1/70 1.43% (FAR)
Plastic mine	2/70 2.86% (FAR)	0/70 0.00% (FAR)	Plastic mine	**54/70** **77.14%** **(Sensitivity)**	11/70 15.71% (FAR)
Other	19/70 27.14% (FAR)	1/70 1.43% (FAR)	Other	8/70 11.43% (FAR)	4/70 5.71% (FAR)
Total FAR:	**23/210 ≈ *10.95%***		**Total FAR:**	**25/210 ≈ *11.90%***	

Table 4 10-fold cross-validation results for HOG+RF (variant II)

Metal detector			Plastic detector		
Object type	Detected as metal mine	Additional side false alarms	Object type	Detected as plastic mine	Additional side false alarms
Metal mine	**66/70** **94.29%** **(Sensitivity)**	0/70 0.00% (FAR)	Metal mine	1/70 1.43% (FAR)	3/70 4.29% (FAR)
Plastic mine	2/70 2.86% (FAR)	0/70 0.00% (FAR)	Plastic mine	**55/70** **78.57%** **(Sensitivity)**	9/70 12.86% (FAR)
Other	17/70 24.29% (FAR)	0/70 0.00% (FAR)	Other	11/70 15.71% (FAR)	2/70 2.86% (FAR)
Total FAR:	**19/210 ≈ *9.05%***		**Total FAR:**	**26/210 ≈ *12.38%***	

Table 5 Time performance of HOG+RF detectors

	Parallelized variant I	**Parallelized variant II**
Detection procedure mean time	4.7 s	21.2 s
Detection procedure mean time per window	0.14 s	0.62 s
Detection procedure mean time per window per classifier's feature	0.64 μs	0.72 μs

improve the results. Obviously, this would also increase time costs (at least without integral imaging). Secondly, one should realize that using more B-scans means in fact taking more advantage of the 3D information and brings the two approaches conceptually closer.

Table 5 reports on time performance for HOG detectors. It demonstrates that though variant I (with 216 features) is faster than our 3D SM approach, the variant II is already much slower, and still uses barely 864 features as opposed to about 1500 used in our approach at detection stage.

In this context, it has not escaped our attention that the HOG-based approach can also be speeded up by integral images. For n_θ bins one would have to introduce $2n_\theta$ integral images cumulating votes (2 accounts for two B-scan orientations). The formulas below demonstrate the idea for the across track case:

$$ii_l(x,t) = \sum_{1 \le j \le x} \sum_{1 \le k \le t} V(j,k,l), \qquad l = 1,\ldots,n_\theta. \tag{19}$$

$$\begin{aligned} H_1(c,l) = {} & ii_l(x_2(c),t_2(c)) - ii_l(x_1(c)-1,t_2(c)) \\ & - ii_l(x_2(c),t_1(c)-1) + ii_l(x_1(c)-1,t_2(c)-1), \end{aligned} \tag{20}$$

where $(x_1(c), t_1(c))$ and $(x_2(c), t_2(c))$ represent the coordinates the cell c spans from and to. Preliminary tests indicated that this idea reduces the total time (variant II) from 21.2 to 2.9 s and the mean time per window × feature to 0.10 μs.

References

1. Abandah G, Anssari N (2009) Novel moment features extraction for recognizing handwritten arabic letters. J Comput Sci 5(3):226–232
2. Azodi H, Zhuge X, Yarovoy A (2011) Balanced antipodal vivaldi antenna with novel transition from feeding line to the flares. In: Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP), pp 1279–1283
3. Bellet P, Leat C (2003) Electrically Small Magnetic GPR Antennas. In: IEEE Antennas and Propagation Society International Symposium, vol 2, pp 235–238
4. Boveiri H (2010) On pattern classification using statistical moments. Int J Signal Process Pattern Recognit 3(4):15–24
5. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
6. Cremer F et al (2003) Feature level fusion of polarimetric infrared and gpr data for landmine detection. In: Proceedings of EUDEM2-SCOT 2003, International Conference on Requirements and Technologies for the Detection, Removal and Neutralization of Landmines and UXO, vol 2, pp 638–642. Vrije Universiteit Brussel, Brussels
7. Cyganek B (2013) Object detection and recognition in digital images: theory and practice. Wiley, New York
8. Friedman J, Hastie T, Tibshirani R (2000) Additive logistic regression: a statistical view of boosting. Ann Stat 28(2):337–407
9. Frigui H et al (2010) Context-dependent multisensor fusion and its application to land mine detection. IEEE Trans Geosci Remote Sens 48(6):2528–2543
10. Giannakis I, Giannopoulos A, Warren C, Davidson N (2015) Numerical modelling and neural networks for landmine detection using ground penetrating radar. In: 2015 8th International Workshop on IEEE Advanced Ground Penetrating Radar (IWAGPR), pp 1–4
11. Jol H (2009) Ground penetrating radar: theory and applications. Elsevier, Oxford
12. Klęsk P, Godziuk A, Kapruziak M, Olech B (2015) Fast analysis of C-scans from ground penetrating radar via 3D Haar-like features with application to landmine detection. IEEE Trans Geosci Remote Sens 53(7):3996–4009
13. Kumar P, Jinhai C, Miklavcic S (2012) Root crown detection using statistics of Zernike moments. In: 12th International Conference on Control Automation Robotics and Vision (ICARCV), pp 1130–1135
14. Manandhar A, Torrione P, Collins L (2015) Multiple-instance hidden Markov model for GPR-based landmine detection. IEEE Trans Geosci Remote Sens 53(4):1737–1745
15. Mease D, Wyner A, Buja A (2007) Boosted classification trees and class probability/quantile estimation. J Mach Learn Res 8:409–439
16. Milisavljević N, Bloch I, Acheroy M (2001) Application of the Randomized Hough Transform to Humanitarian Mine Detection. In: IASTED—7th International Conference on Signal and Image Processing (SIP2001), pp 149–154
17. Missaoui O, Frigui H, Gader P (2011) Land-mine detection with ground-penetrating radar using multistream discrete hidden Markov models. IEEE Trans Geosci Remote Sens 49(6):2080–2099
18. Oyan M, Hamran S, Hanssen L, Berger T, Plettemeier D (2012) Ultrawideband gated step frequency ground-penetrating radar. IEEE Trans Geosci Remote Sens 50(1):212–220
19. Pan X, Zhang X, Lyu S (2012) Exposing image splicing with inconsistent local noise variances. In: IEEE International Conference on Computational Photography (ICCP), pp 1–10
20. Porikli F, Tuzel O (2006) Fast Construction of Covariance Matrices for Arbitrary Size Image Windows. In: IEEE International Conference on Image Processing (ICIP), pp 1581–1584
21. Seyfried D et al (2012) Information Extraction from Ultrawideband Ground Penetrating Radar Data: A Machine Learning Approach. In: 7th German Microwave Conference (GeMiC'2012), pp 1–4
22. Torrione P, Morton K, Collins L (2014) Histogram of oriented gradients for landmine detection in ground-penetrating radar data. IEEE Trans Geosci Remote Sens 32(3):1539–1550
23. Yang H, Roe B, Zhu J (2005) Studies of boosted decision trees for miniboone particle identification. Nucl Instrum Methods Phys Res Sect A Accel Spectrom Detect Assoc Equip 555:370–385. doi:10.1016/j.nima.2005.09.022
24. Yarovoy A (2009) Landmine and unexploded ordnance detection and classification with ground penetrating radar. In: Mol HM (ed) Ground penetrating radar: theory and applications. Elsevier, Oxford, pp 445–478
25. Yarovoy A, Kovalenko V, Fogar L (2003) Impact of ground clutter on buried object detection by Ground Penetrating Radar. In: International Geoscience and Remote Sensing Symposium (IGARSS2003), pp 755–777
26. Zhu Q, Collins L (2005) Application of feature extraction methods for landmine detection using the Wichmann/Niitek ground-penetrating radar. IEEE Trans Geosci Remote Sens 43(1):81–85