DETECTION AND LOCALIZATION OF OBJECTS IN PASSIVE MILLIMETER WAVE IMAGES

Santiago López Tapia*, Rafael Molina*, Nicolás Pérez de la Blanca*
* Dept. of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain
Email: {sltapia, rms, nicolas}@decsai.ugr.es

***Abstract*—Passive Millimeter Wave Images (PMMWI) can be used to detect and localize objects concealed under clothing. Unfortunately, the quality of the acquired images and the unknown position, shape, and size of the hidden objects render difficult this task. In this paper we propose a method that combines image processing and statistical machine learning techniques to solve this localization/detection problem. The proposed approach is used on an image database containing a broad variety of sizes, types, and localizations of hidden objects. Experiments are presented in terms of the true positive and false positive detection rates. Due to its high performance and low computational cost, the proposed method can be used in real time applications.**

***Index Terms*—Millimeter wave imaging, object detection, machine learning, image processing, security.**

I. Introduction

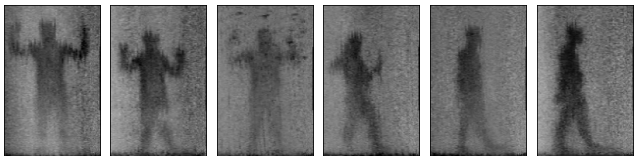
PMMWIs (see fig. 1) can be used to detect objects hidden under clothing. These images are currently being utilized as theft and threat detection systems [1] in places like airports and warehouses. Systems based on PMMWIs should be able to detect concealed objects while, at the same time, incurring in a very low number of false positive detections. Furthermore, undoubtedly, they should also work in real time. Unfortunately, millimeter sensors, and consequently their images, suffer from, among others, the following problems:

1) Low signal to noise ratio.
2) Low resolution, which can be increased by increasing the sampling rate but at the cost of decreasing the signal to noise ratio due to current electronic limitations.
3) Inhomogeneous signal intensity.

Together with the quality of the acquired images, any detection system will have to deal with the unknown position, shape, and size of the hidden objects.

II. Related Work

Previous works have already addressed detection problems using PMMWIs. In [2], K-means is used to segment PMMWIs into three regions: background, body and threats. Unfortunately, the method detects unconnected areas. To solve this problem, the authors use Active Shape Models (ASM) inside the body. However, this approach does not guarantee an adequate segmentation of the subject's body. In [3] Gaussian mixture models are used to characterize image background regions, as well as people with and without threats. Although the reported results are better than those in [2], this method also produces an unconnected body segmentation. In [4] the authors apply noise elimination and then image segmentation using Local Binary Fitting (LBF). The authors use two algorithms for noise removal: Non Local Means (NLM) and Iterative Steering Kernel Regression (ISKR). Although its detection rate is around 90%, its computing time makes it impractical for its use in real-time applications. In addition, its performance significantly decreases when used on noisy or low quality image. In [5] a highly time efficient two-step algorithm, based on denoising and mathematical morphology, was proposed. On noisy or low contrast images it achieves an acceptable detection rate but at the cost of a high false positive detection rate.



(a) (b) (c) (d) (e) (f)

Fig. 1: PMMWI examples. Hidden objects correspond to whiter areas within the body. Unfortunately, not all whiter areas correspond to hidden objects.

In this paper we propose a new methodology to tackle the localization/detection problem on PMMWIs. Our novel approach combines image processing and statistical machine learning techniques to deal with the poor quality of the images and the unknown position, shape, and size of the hidden objects.

The paper is organized as follows. Section III explains the proposed PMMWI processing before detecting hidden objects. Section IV details the proposed feature selection and extraction processes, the used classifiers and the detection procedure. In the experimental section, section V, the used PMMWI database is described and a complete comparison of the features and classification methods utilized provided. Finally, conclusions are presented in section VI.

This paper has been supported by the Spanish Ministry of Economy and Competitiveness under project TIN2013-43880-R, and the European Regional Development Fund (FEDER).

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

III. Image PreProcessing

The acquisition process introduces spatially variant noise, see figs. 2(a)-(d).

The image signal to noise ratio must be increased and the contrast enhanced to be able to detect threats. A combination of linear (local average) and non-linear (local median) smoothing filters is applied. The use of standard convolutional smoothing filters eliminates too much signal due to the noise variability and the small size of some hidden objects. Instead, we have applied statistical filtering, where each pixel value is replaced by the average of a random sample of its neighbouring values. This filter is applied iteratively to obtain an adequate level of smoothing. A 5×5 neighbourhood size was found appropriate empirically. Finally, a 5x5 median filter was used on the output to remove boundary artifacts. The final images are shown in figs. 2(e)-(h).

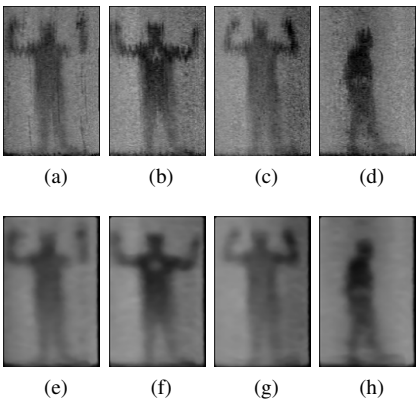


Fig. 2: (a)-(d) show the observed images. (e)-(h) show the processed images.

IV. Learning Object

The proposed object detector is built from a committee of binary classifiers trained using multiscale feature vectors extracted from local patches of the images.

A. Feature extraction

Three different scales have been used to characterize regions containing a hidden object. Given the size of the images in the database, 125×195, and using the size of the largest hidden object in the database, we have selected three sizes (scales): 39×39, 19×19 and 9×9 pixels. To each pixel in the image we associate the three regions of the above sizes centered on the pixel. Only those pixels whose three regions are completely contained within the image (active pixels) are considered. To each active pixel we associate a feature vector constructed concatenating the feature vectors extracted from its three regions. Haar filters [6] and Local Binary Patterns (LBP) codes[7] are used to create the feature vectors.

1) Haar features: It can be observed that the shapes of the hidden objects are very similar to those of some of the Haar filter pattern shown in fig. 3: the filter shown in fig. 3a shares the pattern of the hidden object in fig. 5 row 1 and column 2, whereas the filters in figs. 3b and 3c are similar to areas with the hidden objects in fig. 5 row 1 column 1 and fig. 5 row 1 column 3 respectively. Therefore, a strong response to the filter is expected when applied on the threat. We have used 115 filters on each region resulting in a 3×115 feature vector per active pixel.

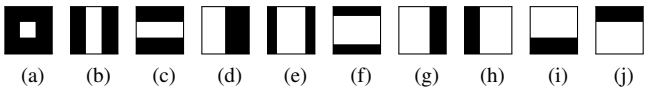


Fig. 3: Examples of Haar filters.

2) LBP features: Local Binary Patterns (LBP)[7] capture image local structures by detecting gray level changes around each pixel. Regions with a strong contrast inside due to the presence of boundaries between hidden object and body or background are highlighted.

We use the extension proposed in [8]; invariant to rotation LBP. For each of the three described above regions centered on an active pixel, the histogram of all LBP configurations obtained using radios in the range 1-4 is built. The feature vector on each pixel is obtained by concatenating the histograms of its three associated regions, it has 261 components.

B. Design of the Classifiers

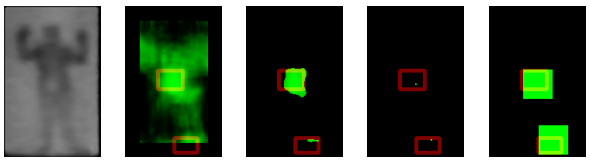
Learning a single classifier is not a good strategy because of the unbalanced sizes of the classes (positive and negative regions). We use a committee of classifiers with a majority vote final criteria. We evaluate six binary classifiers representing different learning approaches: a) Logistic regression with variable transformations; b) Support Vector Machine; c) Random Forest; d) Extreme Random Trees; e) Adaboost.

In our training dataset the number of negative labeled items is around five times the number of positives ones. To overcome a possible overfitting to the negative class different strategies have been used. For AdaBoost we have used an asymmetric boosting[9]. For the other approaches, we split the training set of negative samples into five random partitions, fitting to each of them together with the positive sample partition a different classifier.

C. Detection

The steps to classify a new image are graphically shown in fig. 4. For each active pixel, the corresponding feature vectors are extracted following the training procedure. Then, the probability of the pixel to correspond to a threat is calculated. As a consequence a probability map is produced on the image active pixels (see fig. 4b). Those pixels whose probability are below a prefixed threshold are discarded and assigned to class 0, no threat, (see fig. 4c). The threshold depends on the used classifier. Next, the 2D local maxima of the map

are computed: for all pixels whose 39x39 regions overlap more than an overlapping threshold, we retain the one with the highest probability (non maximum suppression). Then we assign the probability of the local maxima to all pixels inside its 39x39 region. Finally, a list of triplets (coorx, coory, probab) is extracted defining the centers of the regions that potentially contain a hidden object, see fig. 4e.



(a) (b) (c) (d) (e)

Fig. 4: Detection process. The red rectangles indicate the area of the hidden objects.

V. Experimentation

A database of PMMWIs has been created. The database consists of 3309 images of 33 different people. For each person, we took pictures of 12 different objects in 10 body locations: forearm, chest, stomach, thigh, ankle (front), waist (side), armpit (side), arm, ankle (lateral) thigh (lateral), and 2 images without any objects. In summary 463 images with no hidden objects, 2144 containing one hidden object and 702 containing 2 objects. Hidden objects were simulated by bags containing various substances with different millimeter wave responses. Different object sizes were also used. Fig. 5 shows different subjects with hidden objects in different locations.

Hidden objects are surrounded by the smallest box containing it. Visual images taken at the same time as the millimeter ones were used to perform this task. These boxes, as we will explain later, will be used to assess which feature vectors correspond to threats.

1) Training: Training was performed using five-fold cross validation. On each set of samples used, 80% of the feature vectors were used for training and the remaining 20% to estimate the test error.

39×39 pixel regions which completely contain a threat are labelled positive (1), the rest negative (0). Feature vectors which correspond to regions that partially overlap a threat are not included in the training data set.

A five-fold cross validation on the training partition is applied to estimate the hyper-parameters. The same proportions of vectors containing 0, 1, and 2 objects are included in each partition. For each classifier, a subset of the training pixels together with their associated feature vectors are used to find the three classifier hyper-parameter values (see ranges in Table I) with the greatest area under the ROC curve. One pixel out of each 3x3 non-overlapping box in an image is selected and its feature vector considered. Furthermore, for each threat we include at least one pixel whose 39×39 associated box contains it. We now use the three hyper-parameter values on the whole training set to learn three final classifiers and select the one with the greatest area under the ROC curve on the validation set.

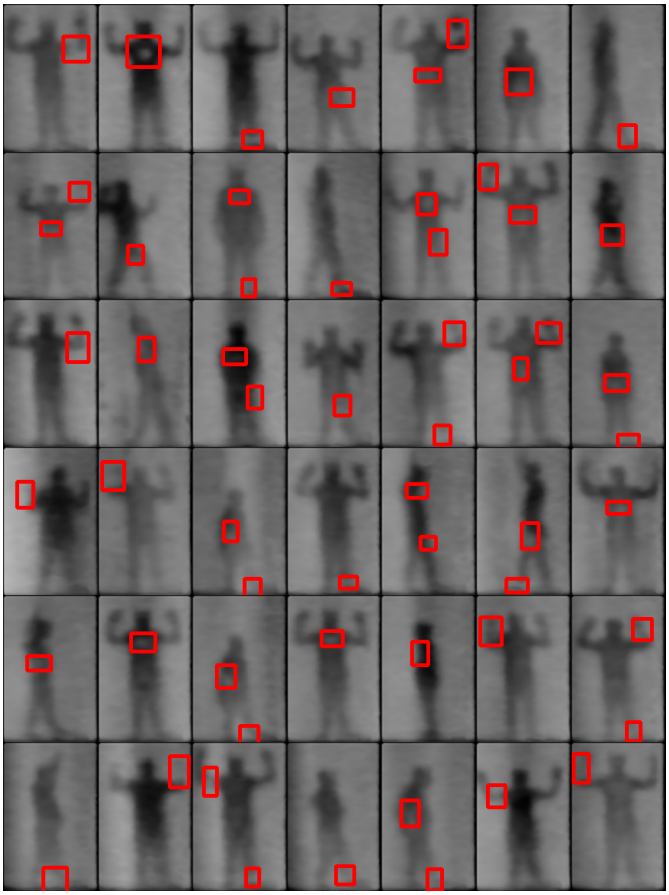


Fig. 5: Examples of PMMWIs with hidden objects. Red boxes indicate object locations. Hidden objects correspond to whiter areas on the body.

Regularized Logistic regression (LR) with quadratic penalty was used as baseline, the regularization parameter (C) was found using adaptive search. We also considered another LR classifier. Using linear and quadratic functions on the features, the LASSO penalty function was utilized to select the relevant features which then were used for LR with quadratic penalty. For Radial Basis Support Vector Machine (SVM), Random Forest (RF) and Extreme Random Trees (ERT) a grid search is used to estimate the hyper-parameters. Feature are normalized by mean and variance before training.

Table I shows the hyper-parameter ranges used for each classifier.

Each classifier output is used to detect threats. Its associated ROC curve is used to calculate a threshold, normalized to the interval $[0, 1]$. Pixels whose outputs are above the threshold are labelled as threat (label 1). Notice that, as we have already indicated, an overlapping threshold is used to perform non maximum suppression.

2) Evaluation: The reported test error is the mean of the five cross validation errors. We consider that a pixel (and its associated box) correctly detects a threat if the box overlaps

Classifier	Parameter	Range
RL	C	[0, 5]
RL quadratic	C	[0, 5]
SVM	C	[0.1, 100000]
	γ	[0.00001, 10]
Random Forest	Trees	[100, 300]
	N° features	[8, 30]
	Min examples per leaf	[1, 50]
Extreme Random Forest	Trees	[100, 300]
	N° features	[8, 30]
	Min examples per leaf	[1, 50]

TABLE I: Hyper-parameters grid for each classifier.

the threat by at least 50%.

3) Software: The libraries *OpenCV* [10], *scikit* [11], *scikit-image* [12] and our own implementation of the Haar filters have been used in the experiments. For grid search the python package *optunity* [13] was used.

4) Results: The results obtained using Haar and LPB features are shown in table II. The best thresholds for threat detection and also the best overlapping thresholds for non maximum suppression used during training are also included. For RF and ERT we used parallel implementations ran on 16 threads. The computation times, using an Intel Xeon E5-2630 v3 to 2.40GHz with 8 cores, during testing are reported in Table III. Notice that they are close to real time for a threat detection application.

Results of experiments for Haar features				
Classifier	TP	FP	Threshold	Overlap
RL	0.84±0.018	10.56±10.78	0.6	0.5
RL quad	0.91±0.011	8.76±9.11	0.7	0.5
SVM	0.92±0.030	6.51±6.60	0.75	0.3
RF	**0.94±0.0093**	**4.03±3.82**	**0.7**	**0.3**
ERT	0.94±0.006	5.04±5.00	0.7	0.5
Boosting	0.93±0.011	6.43±6.28	0.5005	0.3
Results of experiments for LBP features				
Classifier	TP	FP	Threshold	Overlap
RL	0.92±0.0084	10.24±10.54	0.7	0.5
RL quad	0.92±0.010	10.27±10.57	0.7	0.5
SVM	0.93±0.014	8.82±8.90	0.6	0.4
RF	**0.92±0.0052**	**6.62±6.53**	**0.55**	**0.3**
ERT	0.90±0.012	6.11±6.03	0.55	0.3
Boosting	0.92±0.012	9.99±10.20	0.5004	0.5

TABLE II: Experimental results. The second column shows the rate of true positive (TP) detected threats together with the standard deviation. The third column shows the mean and standard deviation of false positive (FP) detected pixels in each image for the thresholds and overlaps in the fourth and fifth columns.

Although the performance of almost all models is similar, RF and ERT show the best figure of merits when used with Haar features. A key point is the FP average per image. All models perform well since the number of false positives is always below 10% when a 100% TP detection is imposed on the system. In fig. 6 we show the trade-off between true positives and true negatives when we classify new images. The crossing point of both curves defines the capacity of the system to classify new images.

	Haar		LBP	
Classifier	Total	Per image	Total	Per image
RL	302.77	0.0915	4939.65	1.4927
RL quadratic	341.70	0.103	4939.69	1.4928
SVM	1277.04	0.3859	6907.10	1.8758
RF	**740.00**	**0.2236**	**5497.47**	**1.6619**
ERT	1202.60	0.3634	5507.93	1.6645
Boosting	1965.54	0.5940	7006.23	2.1173

TABLE III: Testing Total (all the images) and per image times in seconds for each method.

For the best combination, RF and Haar features, fig. 7 shows: (a) the histogram of false positives per image; (b) the position of the first true positive in the list of detected locations in the image, ordered by decreasing probability; and (c) the mean probabilities of the four highest probabilities per image. Fig. 7a shows that the mode of the mean number of false positives per image is 4. Fig. 7b indicates that the vast majority of true positives are among the first two detected regions, and fig. 7c the tendency of the classifier to assign the highest probabilities to regions overlapping true threats, see also fig. 7b, which justifies the use of a high threshold for threat detection.

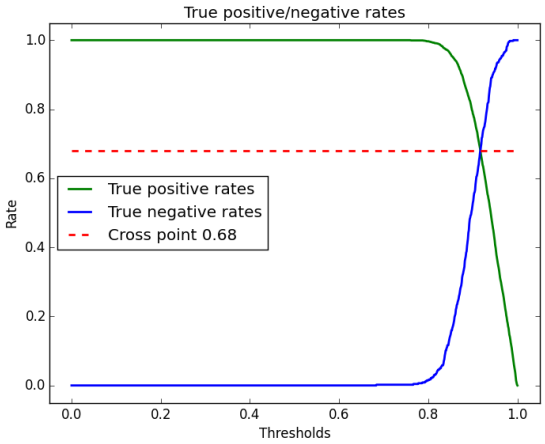


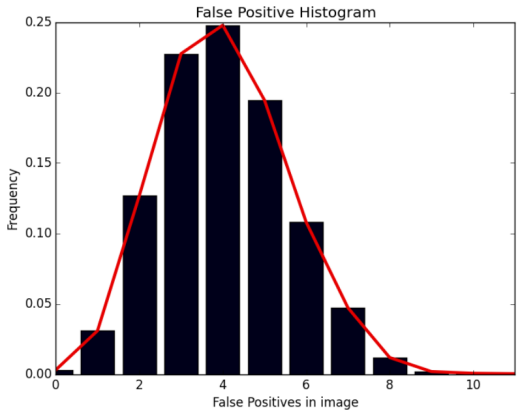
Fig. 6: The curves show the accuracy of the model on new images for a range of detection probability thresholds. The cross point is at the 68% of accuracy for both classes.

VI. Conclusions

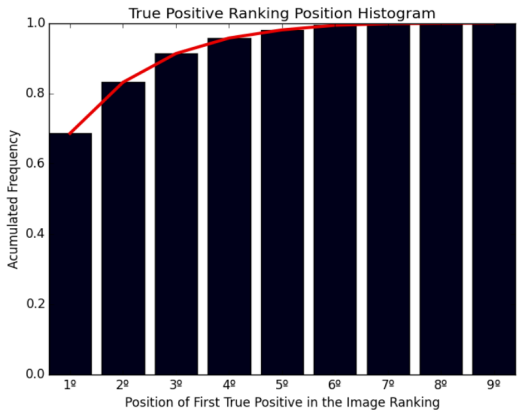
In this paper we have proposed a new method to solve localization/detection problems using PMMWIs. The method combines image processing and statistical machine learning techniques and performs very well even on very noisy and poor quality images. The experiments carried out also indicate that Haar filters provide very good features for this classification problem. To the best of our knowledge, the image database used is the largest, and with the greatest variety of object types and sizes, ever used for this problem.

VII. Acknowledgements

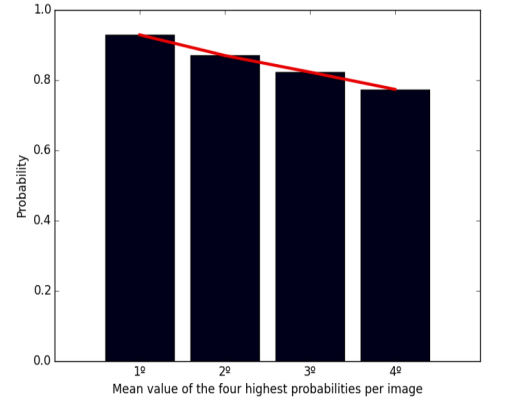
We would like to acknowledge the use of the real PMMWIs provided by Wavecamm (http://wavecamm.com).



(a)



(b)



(c)

Fig. 7: Histograms of behaviour of the best model.

REFERENCES

[1] N. E. Alexander, C. Callejero Andrés, and R. Gonzalo, "Multispectral mm-wave imaging: materials and images," in *Passive Millimeter-Wave Imaging Technology XI*, vol. 6948, Apr. 2008, p. 694803.

[2] C. Haworth, B. Gonzalez, M. Tomsin, R. Appleby, P. Coward, A. Harvey, K. Lebart, Y. Petillot, and E. Trucco, "Image analysis for object detection in millimetre-wave images," in *Passive Millimetre-wave and terahertz imaginh and technology*, vol. 5619, 2004, pp. 117–128.

[3] C. Haworth, Y. Petillot, and E. Trucco, "Image processing techniques for metallic object detection with millimetre-wave images," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1843 – 1851, 2006.

[4] O. Martínez, L. Ferraz, X. Binefa, I. Gómez, and C. Dorronsoro, "Concealed object detection and segmentation over millimetric waves images," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, June 2010, pp. 31–37.

[5] I. G. Maqueda, N. P. de la Blanca, R. Molina, and A. Katsaggelos, "Fast millimetre wave threat detection algorithm," in *European Signal Processing Conference (EUSIPCO 2015)*. Nice (France), September 2015.

[6] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Sixth International Conference on Computer Vision*, Jan 1998, pp. 555–562.

[7] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, pp. 51–59, Jan. 1996.

[8] T. Ahonen, J. Matas, C. He, and M. Pietikäinen, "Rotation invariant image description with local binary pattern histogram fourier features," in *Image Analysis*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5575, pp. 61–70.

[9] P. Viola and M. Jones, "Fast and robust classification using asymmetric adaboost and a detector cascade," in *Advances in Neural Information Processing System 14*. MIT Press, 2001, pp. 1311–1318.

[10] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. a. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 6 2014.

[13] M. Claesen, J. Simm, D. Popovic, and B. D. Moor, "Hyperparameter tuning in python using optunity," in *International Workshop on Technical Computing for Machine Learning and Mathematical Engineering (TCMM 2014)*, Leuven, Belgium, 2014 2014.