2016 12th World Congress on Intelligent Control and Automation (WCICA)
June 12-15, 2016, Guilin, China

An Improved ORB, Gravity-ORB for Target Detection on Mobile Devices

Zhuqing Hu and Yongshi Jiang

***Abstract*— Feature matching is at the base of the target detection problem. Current methods rely on costly descriptors for detection and matching. This paper presents an improved feature descriptor based on ORB, called Gravity-ORB for target detection in mobile devices. Compared with traditional descriptors such as SIFT or ORB, the concept design can perform fast feature matching under the condition of keeping the restriction on robustness, even in the case where mobile devices have limited computational capacity. Specially, Gravity-ORB reduces the complexity of feature computation for mobile devices with less computing by using gravity acceleration sensors. In the end, experiments conducted in smart phones and tablets demonstrate the effectiveness and real-time performance of the proposed method.**

I. Introduction

Vision-based fast and robust target detection in varying orientation and occlusion levels is the center problem of many computer vision applications. In many applications, especially under mobile settings, a major problem is that the computational resources are limited. This motivates the usage of lightweight features to model the target. One approach for detecting object is to extract feature firstly and then recognize according to these discriminating features. Common methods for feature point matching are BRIEF [1], FREAK [2], BRISK [3], and ORB [4]. These features allow reliable matching of images at acceptant computational cost, with binary descriptors are the core. Matching with binary operators can be computed very efficiently on modern CPU. However, nowadays it is a big data era, and there are a large number of data which is needed to be processing at every moment. Target detection with these features on mobile is not efficient which need more memory consumption and resource usage. The response time of detection is intolerable and existing detection methods are inadequate to model the real environment problem.

Despite huge opportunities, implementing an efficient and accurate algorithm for object detection on mobile phone is still an open challenge. State-of-the art object recognition methods had been successfully applied to object detection and obtained reasonable recognition rates [5]-[7]. The algorithm based multiscale detection schemes was proposed by using eight detection models for each half octave scales [8]. The image features had to be computed only once each half octave and there was no need for feature approximation. Moreover, A. Alahi proposed one method that Features extracted by the master camera were used to detect the object of interest in the slave camera without the use of any training data [9]. Target detection could be done well based on these methods, but they all required significant computation and memory. With limited mobile resources, D. Kartashov introduced a new artificial landmark design and fast algorithm for detecting and tracking them in arbitrary images and they can performance well for different hardware platform [10]. Besides, a weight vectors of linear was compressed by Y. Kawano, which led only slight performance loss [11]. Efficient target detection were formed at some specific application scenarios with a suitable cost of mobile memory. S. S. Kumar proposed an approach to multi-frame object detection using Hough Forests [6], and demonstrated that multi-frame generalization of [6] improved the detection performance [12]. The Gaussian elimination algorithm is built to estimation holography, and this algorithm was integrated into a fast target recognition framework based on trained BRIEF features on mobile devices [13]. LLDB directly computed a binary string for an image patch using simple intensity and gradient difference tests on pairwise grid cells within the patch [14]. However, the time of response was not satisfied with mobile environment by using these improved features.

Among the mobile AR systems, one of the key steps is object detection. The model of server-client scheme was used in [15][16], and they used local image features to identify objects and return response to the client. Whereas, there were two parts of the system, and more times were needed to data transmission. Mobile also has their unique characteristics, and built-in inertial sensors of mobile phone were demonstrated to increase efficiency for accurate blur estimation [17]-[20]. Therefore, the mobile sensor can be considered to optimize the target detection. In this paper, we focus on the static planar target detection on the mobile platform and analyze the process of feature detection with the mobile unique characteristic. The indoor environment is the main test scene, and the target detection will be examined with different illumination intensity and viewpoint. The gravity acceleration sensor is used to optimize the process of feature detection, and tests show that the optimized approach can improve the detection rate. More specifically, objects are successfully detected even if the cameras have a meaningful change of image quality, illumination, and viewing point. Fig. 1 shows an example of this procedure.

In the remainder of this paper, the ORB algorithm are described in section II, which is the foundation of our method. The Gravity-ORB method is elaborated in section III. Some tests results which show its performance are further

Zhuqing Hu and Yongshi Jiang are with the Department of Integrated Information System Research Center, Institute of Automation Chinese Academy of Sciences, Beijing (email: {huzhuqing2013, yongshi.jiang}@ia.ac.cn).

978-1-4673-8414-8/16/$31.00 ©2016 IEEE

Fig. 1. Target detection example

provided in Section IV. Finally, the conclusion is presented in section V.

II. Related Works

For target detection and matching, robust features with scale-invariant and rotation-invariant properties are essential. High level features are normally more unique but difficult to identify. The challenge lies in finding of scale and rotation-invariant features, as well as making a trade-off between computational efficiency and feature robustness.

The ORB(oriented FAST and rotated BRIEF) descriptor was initially proposed by Ethan Rublee [4]. As indicated by the name, ORB feature is the combination of FAST detector and oriented BRIEF feature. For each frame, a 3-level Gaussian pyramid is built and FAST algorithm is used to select points at each level of the Gaussian pyramid. For first N points, we first employ the FAST corner keypoint detector to get more than N keypoints by setting the threshold low enough. The first N points are chosen as the key points. The intensity centroid of a patch of fixed size around the key point is calculated to find the orientation of the keypoint with respect to the origin, as is shown in Fig. 2, by introducing the geometric moments.

The moments m_{pq} of an image patch R are defined as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y), (x,y) \in R \tag{1}$$

where $T(x,y)$ is the intensity at point $(x,y$. And with these moments, the centroid P is shown in function (2).

$$P = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{2}$$

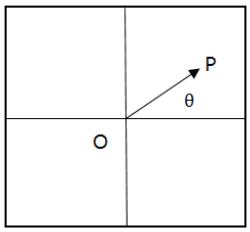

Fig. 2. Intensity centroid model

So the orientation of the FAST keypoint becomes:

$$\theta = a \tan 2 \left(m_{01}, m_{10} \right) \tag{3}$$

When the θ is available, we can employ the Harris corner measure to order the FAST keypoints, and then pick the top N points.

By rotating the major axis in different angular steps, we can use rotated BRIEF. The BRIEF feature is 256-element vector, which can be computed using simply intensity difference tests. Each bit of the descriptor comes from making a binary comparison of every two randomly selected pixels.

More specifically, a binary test on an image patch p of size SxS is defined as:

$$\tau \left(p : x, y \right) = \begin{cases} 1 : p(x) < p(y) \\ 0 : p(x) \geq p(y) \end{cases} \tag{4}$$

where $p(x)$ and $p(y)$ are the intensity of p at point x. With an unique chosen set of (x,y) pairs, the BRIEF descriptor is generated as a string of n bits:

$$f_n \left(p \right) = \sum_{1 \leq i \leq n} 2^{i-1} \tau \left(p : x, y \right) \tag{5}$$

As the ORB descriptors are binary vectors, Hamming distance can be used to measure whether the two features are matched or not. The detection of ORB features goes in following steps:

1) Construct the 3-level pyramid of original image based on a 5x5 Gaussian kernel;
2) Use the FAST detector to find the corner points at each level of pyramid image;
3) The first N points, which are ordered according to their Harris corner response, are selected as the key point.
4) Calculate the intensity centroid of a circular region around key points, and compute the direction of the BRIEF feature.
5) Rotate the BRIEF feature with a fixed angle around direction and build a rotated BRIEF features;

Compared to SURF or SIFT, the ORB feature is much faster, and it is more suitable for the mobile environment.

III. Gravity-ORB

The gravity-ORB presented in this paper consists of two parts. The main design principles of those two parts introduced briefly in this section.

A. Mobile coordinate system

In this paper, gravity acceleration sensor is used to calculate ORB feature, which makes the computational process more efficiency. To add the gravity information to the image, we must know about the coordinate of the mobile. The origin of the mobile coordinate system is at the top right of the screen, and the arrow is pointing to positive (Fig. 3).

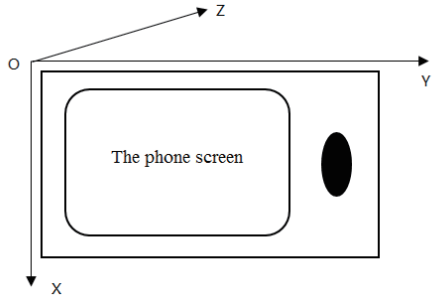

Fig. 3. Mobile coordinate system

From -1 to 1, with floating point unit for grades, seeing the following scenario, the phone screen up at the right time (z-axis toward the ground) horizontally, (x, y, z) is denoted by (0, 0, 1). (x, y, z) is denoted by (0, 0, -1), with the phone screen down at the right time (z-axis overturned) horizontally, . When the phone screen is put x-axis in the air, (x, y, z) is denoted by (1, 0, 0). When the phone screen is put y-axis upwards, (x, y, z) is denoted by (0, -1, 0). Using x, y, z three values to seek trigonometric functions, it can accurately detect the motion of the mobile phone.

B. Gravity-ORB Method

A 2D point is indicated by $m = [u, v]^T$. A 3D point is indicated by M = $[X, Y, Z]^T$. We use $\widetilde{x}$ to denote the augmented vector by adding 1 as the last element: $\widetilde{x} = [u, v, 1]^T$ and $\widetilde{x} = [X, Y, Z, 1]^T$. A camera is modeled by the usual pinhole: the relationship between a 3D point M and its image projection m is given by

$$s\widetilde{m} = A\,[R \quad t]\,\widetilde{M} \qquad withA = \begin{bmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$R = R_x R_y R_z \tag{7}$$

where s is an arbitrary scale factor; (R, t), called the extrinsic parameters, is the rotation and translation which relates the world coordinate system to the camera coordinate system; A is called the camera intrinsic matrix. (u_0, v_0) are the coordinates of the principal point, α and β are the scale factors in image u and v axes, and c is the parameter of describing the skewness of the two image axes.

To acquire the direction of gravity which is added to the image, we first need to know when phone gesture is taken. Therefore, the calculation of handset rotation angle is one of the key steps for the algorithm. The gravity sensor system is used to complete the task of computing the angle. R_x, R_y, R_z are rotation matrix around X, Y, Z, which is shown the x axis, y axis and z axis in Fig. 4. The orientation of key points are obtained from the gravitational component of the gravity sensor which gain the rotation matrix (See function (7)). As is shown in Fig. 4, (a, b, c) is the gravitational component, where $a, b, c \in (-1, 1)$.

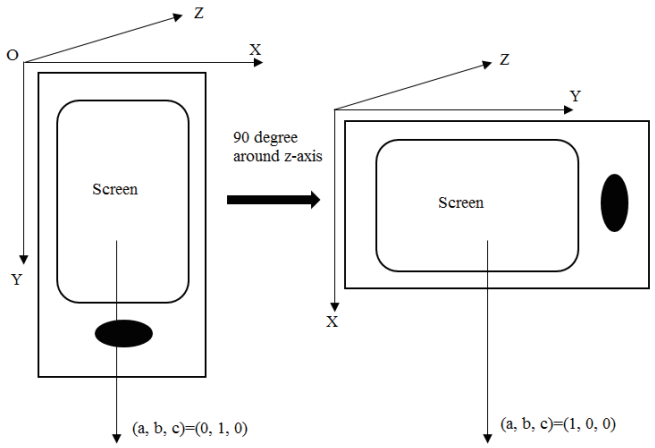

Fig. 4. Rotation schematic

Main direction is the cumulative sum of gradient with the neighborhood pixels. The process of orientation computation go through a lot of reduplicated steps, which increases the algorithm complexity. Instead of orientation of rotated BRIEF, the gravity direction as the direction of descriptor can reduce the computational complexity.

When using mobile devices to shoot the scene, due to the human factor, the phone screen cannot be completely parallel to the object. The different degree of tilt are produced. The degree of tilt can be obtained by gravity acceleration sensor, which are converted into a rotation angle in different directions. We describe the gravity-ORB as illustrated in Algorithm 1.

Gravity alignment descriptor can be described in detail as follows:

1) The standard gravity vector G_1 is that put phone screen y-axis downwards. We recode the (x_g, y_g, z_g) to be (0, 1, 0).
2) When we capture the scene in real time, we recode another gravity vector G_2.
3) G_1 and G_2 will be subtracted to give a new gravity vector G= (x', y', z'). We can get the rotation angle θ from G.
4) At last, the descriptor will be rotated with the rotation angle θ , and then aligned with the descriptor. (Fig. 5)

When detecting feature point, ORB will calculate direction angle for each feature point, so that each feature point has one angle property. In the calculation of feature point descriptor, the point will be rotated with this angle. Traditional ORB feature requires computing the sine and cosine of the angle,

Algorithm 1 the Gravity-ORB

initial gravity vector G_1.
while (1) **do**
  Image capture.
  Initial orientation θ.
  Get gravity vector G_2 based on Gravity Acceleration sensor.
  if G_1 is not equal with G_2 **then**
    Change θ.
  end if
  Construct the 3-level pyramid of image based on Gaussian kernel.
  Use the FAST detector to find the corner point at the pyramid image.
  get the first N points, according to the Harris corner response.
  Use θ to replace the direction of the BRIEF feature.
  Rotate the BRIEF feature.
  Apply PCA to each of the local feature.
  Use Hamming distance to judge if the two feature are matched or not.
end while

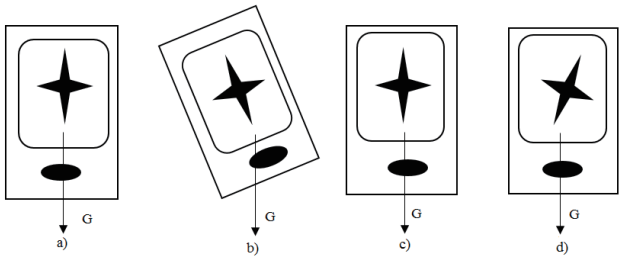

Fig. 5. Gravity alignment descriptor

and the calculation of the gradients are needed the implementation of the five redundantly multiplication. Although the calculation time of a multiplication overhead is small, a huge cost will be produced if there is a large number of data to calculated.

The Gravity-ORB is efficiency and the advantages are as follows: 1) The Gravity-ORB will not compute direction angle when detecting feature point. 2) The angle calculated by the gravity vector will be passed to the ORB computing descriptor. 3) The angle from step 2) will be used to rotate the point early when computing the descriptor. Because the gravity direction is the main direction, we only need one time interval. 4) We simplify the process of computing descriptor, and reduce the number of multiplications.

The Gravity-ORB method improves the original algorithm for calculation direction angle and speed up the process of extracting feature points. Compared with the original algorithm, Gravity-ORB only use gravity vector to calculate the direction angle, which is more efficient than the ORB feature. In addition, gravity-ORB only calculate sin and cosine once instead of the original ORB feature, which needs calculate for each feature points. Besides, Gravity-ORB need one time multiplication, while the original ORB feature needs five times.

IV. Experimental Results

To show the effectiveness and efficiency of the proposed Gravity-ORB, the following two tests were performed. For the test, we build a system with Eclipse and Android-OPENCV-SDK on Windows 7, which run on Android 4.4.

In the test, the inputs are the number of detecting points, which are set to be $num = i$,$(i \in (50, 500))$. The time of detecting points by Gravity-ORB are set to be $Gtime$, and all recodes are in milliseconds. Fig. 6 shows the result of the comparison between the proposed approach and other methods. We can see from the result that gravity-ORB reduces the time of computing. That means by using the Gravity-ORB, the more data are handled with the the same time. And with the increase of the points, the advantages is more obvious. It is efficient for the mobile devices.

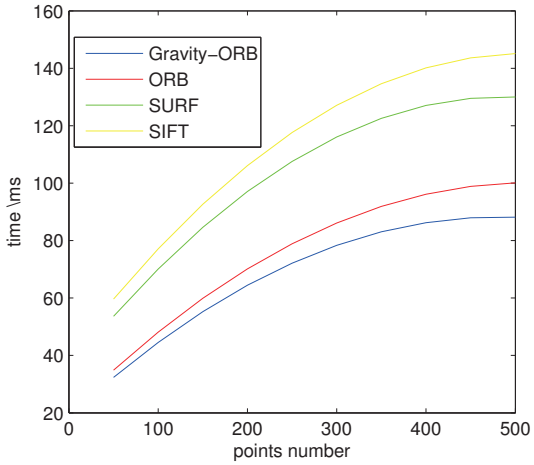

Fig. 6. Computing time for the different descriptors

In order to test the object detection accuracy of the method, the following experiment are shown the results. The experiments are designed in two ways, which are the target detection with different viewpoint and illustrate. The result presents the performance of the detection and matching of the approach presented in this paper. The experiment shows that gravity-ORB has a good performance of different viewpoints, and the target is located by the algorithm accurately. (See Fig. 7 and Fig. 8.)

Besides, the illumination was examined with gravity-ORB. From Fig. 9 and Fig. 10, the accuracy of target detection was influenced on the variation of light intensity. when illumination changed much, the corner of image drift away a lot. The results shows that the brightness is a key problem of the target detection.

V. Conclusions

This paper addresses the problem of target detection on smartphones. In detecting the target object, feature points

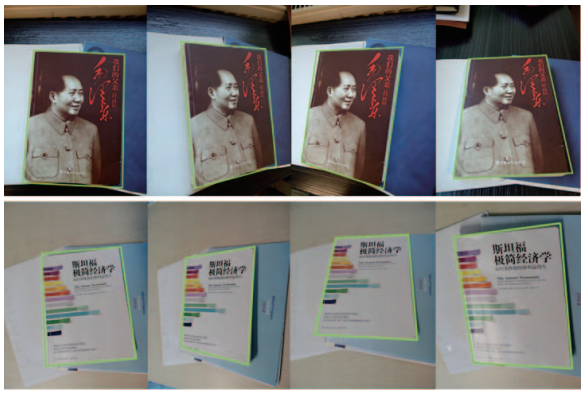

Fig. 7. Viewpoint change test: first row is detecting at the standard of light about different viewpoint, and second row shows that target was detected under lamplight at night.

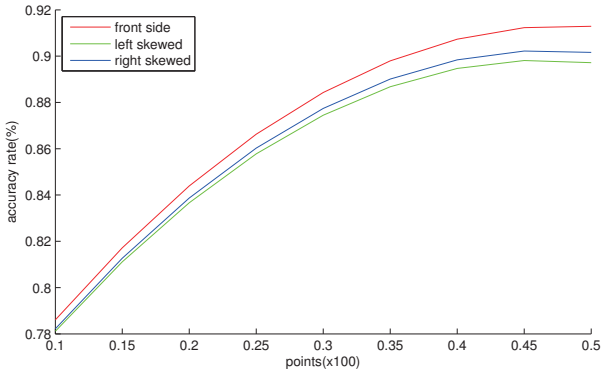

Fig. 8. Gravity-ORB of target detection with different viewpoint

and descriptor extraction from images have two significant drawbacks: CPU consumption and weak robustness depending on environment conditions. In this paper fast and robust solution for target detection is presented which can run in real time conditions even on mobile devices with limited computational power. We consider the mobile sensor for target detection, and the proposed procedure has been experimentally validated on a mobile device by using the gravity acceleration sensor in the presence of optimizing the ORB feature. The experiment shows that this improved approach significantly speed-up the target detection. It considerably reduces the computational load in real-time implementation on mobile devices.

With this method, the proposed target detection method with gravity acceleration sensor outperforms those without any sensors. But, the accuracy of target detection is still affected by the bright, and we should consider about the change of the lighting at future. Furthermore, the other way to improve the accuracy should be taken into account; for instance, not all objects need to be retrieved at every detecting time. Besides, we can also enlarge the test database to make the result more accuracy and add location into the system so as to decline the retrieving scope.

Fig. 9. Illumination change test: first column is detecting at the standard of light and right last column shows that target was detected under lamplight at night. second column is detected at dark lighting under lamplight, and high brightness was used at column three.

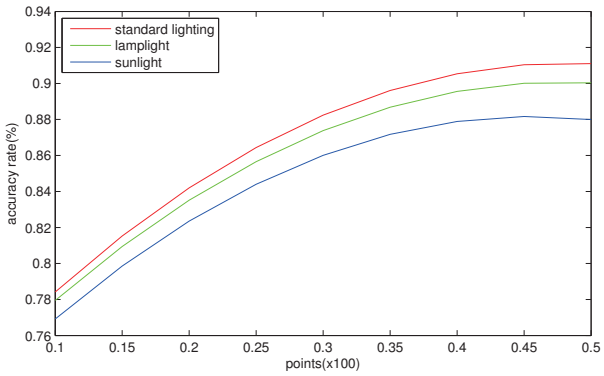

Fig. 10. Gravity-ORB of target detection with different illumination.

REFERENCES

[1] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, Brief:Binary robust independent elementary features, *Computer VisionECCV*, vol. 6314, pp. 778–792, 2010.

[2] A. Alahi, R. Ortiz, and P. Vandergheynst, Freak: Fast retina keypoint, *Computer Vision and Pattern Recognition*, pp. 510-517, 2012.

[3] S. Leutenegger, M. Chli, and R. Siegwart, Brisk: Binary robust invariant scalable keypoints, *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2548-2555, 2011.

[4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, ORB: An efficient alternative to sift or surf, *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564-2571, 2011.

[5] P. F. Felzenszwalb, R. B. Girshick, and D. Mcallester, Cascade object detection with deformable part models, *Computer vision and pattern recognition*, pp. 2241–2248, 2010.

[6] J. Gall and V. Lempitsky, Class-specific hough forests for object detection, *Decision forests for computer vision and medical image analysis*, pp. 143–157, 2013.

[7] K. Kungcharoen, P. Palangsantikul, W. Premchaiswadi, Development of Object Detection Software for a Mobile Robot using an AForce.Net Framework, *International Conference on ICT and Knowledge Engineering*, pp. 201–206, 2011.

[8] A. D. Costea, A. V. Vesa, S. Nedevschi, Fast Pedestrian Detection for Mobile Devices, *Intelligent Transportation Systems*, pp. 2364–2369, 2015.

[9] A. Alahi, D. Marimon, M. Bierlaire, M. Kunt, A Master-Slave Approach for Object Detection and Matching with Fixed and Mobile Cameras, *IEEE International Conference on Image Processing*, pp. 1712-1715, 2008.

[10] D. Kartashov, A. Huletski, K. Krinkin, Fast Artificial Landmark Detection for Indoor Mobile Robots, *Computer Science and Information Systems*, vol. 5, pp. 209-214, 2015.

[11] Y. Kawano and K. Yanai, Mobile Object Detection through Client-Server based Vote Transfer, *Computer Vision and Pattern Recognition*, pp. 3290–3297, 2014.

[12] S. S. Kumar, Mobile Object Detection through Client-Server based Vote Transfer, *Computer Vision and Pattern Recognition*, pp. 3290–3297, 2012.

[13] O. Blianiuk, Fast Target Recognition on Mobile Devices: Revisiting Gaussian Elimination for the the Estimation of Planar Homograph, *Computer vision and pattern recognition workshops*, pp. 119–125, 2014.

[14] Xin Yang, Learning Optimized Local Difference Binaries for Scalable Augmented Reality on Mobile Devices, *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no.6, pp. 852–865, JUNE 2014.

[15] S. Gammeter, A. Gassmann, L. Bossard, Server-side object recognition and client-side object tracking for mobile augmented reality, *Computer Vision and Pattern Recognition Workshops*, pp. 1–8, 2010.

[16] S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. I know what you did last summer: object level auto-annotation of holiday snaps, *Computer Vision, IEEE International Conference on*, pp. 614–621, 2009.

[17] F. Sroubek, P. Milanfar, Space-variant image deblurring on smartphones using inertial sensors, *Computer Vision and Pattern Recognition Workshops*, pp. 191–192, 2014.

[18] Yue Li, D. K. Jha, A. Ray, T. A. Wettergren, Feature Level Sensor Fusion for Target Detection in Dynamic Environments, *American Control Conference*, pp. 2433–2438, 2015.

[19] G. Wittenburg, N. Dziengel, S. Adler, Z. Kasmi, M. Ziegert, J. Schiller, Cooperative event detection in wireless sensor networks, *Communications Magazine, IEEE*, vol. 50, pp. 124-131, December 2012.

[20] N. Katenka, E. Levina, and G. Michailidis, Local vote decision fusion for target detection in wireless sensor networks, *Signal Processing, IEEE*, vol. 56, pp. 329-338, Jan 2008.