DISTRIBUTED SOURCE CODING: APPLICATION IN BIOMETRICS

Maurício Ramalho[1], Sanchit Singh[1,2], Paulo Lobato Correia[1,2], Luís Ducla Soares[1,3]

[1]Instituto de Telecomunicações, [2]Instituto Superior Técnico, [3]Instituto Universitário de Lisboa (ISCTE-IUL)
Torre Norte - Piso 10, Av. Rovisco Pais, 1, 1049-001, Lisboa, Portugal
phone: + (351) 218418461, fax: + (351) 218418472, email: {mar, sanchit, plc, lds}@lx.it.pt

ABSTRACT

Distributed source coding (DSC) applications range from sensor networks to video coding. In this paper, DSC principles are integrated in a biometric system to provide secure template storage. The proposed biometric recognition system uses the palmprint as a biometric trait together with a state-of-the-art feature extraction technique that produces binary templates. The hand images are captured in an unconstrained way, i.e., the user can place his hand freely within the camera's field of view and the background is not required to be constant. The proposed system was tested in indoor environments and promising recognition accuracy results were achieved, when performing identification on a small database, acquired using the proposed system.

Index Terms— distributed source coding, secure biometrics, LDPC, unconstrained palmprint, palmprint recognition

1. INTRODUCTION

Biometric recognition systems are becoming powerful means for automatic personal recognition. Some physiological (e.g., iris, fingerprint, face, palmprint) or behavioral characteristics (e.g., keystroke dynamics, signature, gait) can be used as biometric traits [1].

Early palmprint studies were reported in 1998 [2] and since then it has been accepted as a good candidate for biometric recognition. Most of the reported research work on palmprint recognition systems requires pegs [3], contact-based image acquisition devices (e.g., optical scanner) [4,5,6,7] or contactless image acquisition devices (e.g., camera) with a uniform background [8,9].

More recently, efforts to recognize palmprint images captured in unconstrained scenarios (e.g., non-uniform background, hand captured with different perspectives) are under way. This is an active research area and not much work has been reported. Han et al. [10] proposed the first contactless palmprint recognition system operating in unconstrained environments. In [10], two webcams were placed in parallel, one capturing near-infrared (NIR) hand images, while the other captures in the visible part of the spectrum. The NIR image was used to segment the hand from the background and, subsequently, to determine the hand's location in the RGB image. However, the approach in [10] required the hand to be placed in a vertical position, allowing a maximum variation of -15 to +15 degrees. Additionally, the system setup required modifying one of the webcams to capture NIR images and a precise alignment between the two cameras had to be performed.

In this paper, an unconstrained palmprint identification system based on a single webcam is proposed. Since the background is not uniform, background subtraction is combined with a fast skin color detection algorithm to segment candidate hand regions from acquired images. A background update algorithm is also implemented to timely incorporate the changes observed in the acquired images into the background model.

Biometric recognition systems should preserve their users' privacy by storing data in a non-invertible way. In fact, biometric data is very sensitive because a person cannot change a biometric trait if it is somehow compromised.

Existing biometric recognition schemes with secure template storage, typically use an error-correcting code (ECC) to handle intra-user variations (or acquisition noise). Davida et al. [11] discussed that error correction could be applied in two different stages: acquisition or verification. One application of ECC at verification stage was proposed by Juels and Wattenberg [12], the fuzzy commitment scheme, where a hash function is also used to ensure biometric data's privacy. Another application was proposed by Juels and Sudan [13], the fuzzy vault scheme, which consists in hiding the biometric template under a set of polynomial coefficients. The probe biometric template is then used to recover the coefficients using a Reed-Solomon code. More recently, Nagar et al. [14] proposed a technique that allows secure template storage based on a Low-Density Parity-Check (LDPC) code and a cryptographic hash function (CHF), showing how distributed source coding principles can be applied to biometric systems.

In the proposed system, distributed source coding principles are used as follows. Given two correlated sources (two biometric templates from the same user, b and b'), the goal is to recover b by using b' and a set of parity bits that were extracted from b. This strategy is combined with a CHF that is used to achieve secure template storage. The remainder of the paper is organized as follows. The proposed system architecture is presented in Section 2, which also describes the background subtraction (2.1), skin color detection (2.2), hand detection and ROI extraction (2.3), background update (2.4), feature extraction (2.5), secure template storage (2.6)

The authors acknowledge the support of Fundação para a Ciência e Tecnologia (FCT) under project PTDC/EEA-TEL/098755/2008.

and secure template matching (2.7). In Section 3, experimental results are presented and, finally, conclusions are drawn in Section 4.

2. SYSTEM ARCHITECTURE

The proposed system architecture is illustrated in Figure 1.

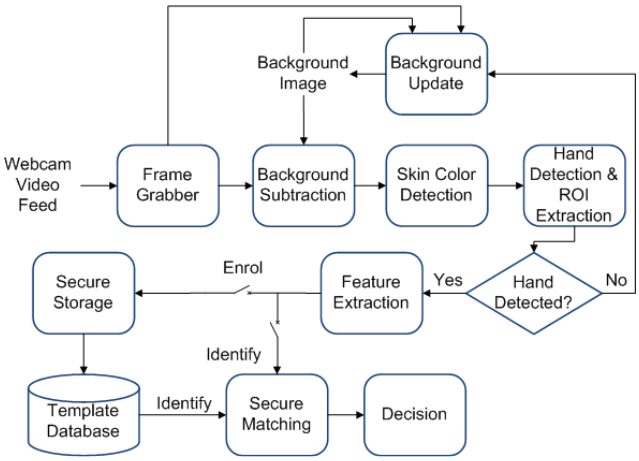


Figure 1 – Proposed system architecture.

2.1. Background Subtraction

The hand region is segmented using a background subtraction technique, in the YCbCr color space. Given a previously stored background image, *B*, (Figure 2 (a)) and any captured frame, *I*, (Figure 2 (b)), the background subtraction is defined as

$$BS = |I - B|. \tag{1}$$

To make the background subtraction more robust to illumination variations, the luminance values stored in the Y component are discarded and only the Cb and Cr values are used. Hence, the binary foreground mask, *FM*, is computed as:

$$FM = (BS_{Cb} + BS_{Cr}) \geq \beta, \tag{2}$$

where β is a threshold, determined experimentally. For the used camera setup, $\beta = 10$ was found to produce good results. It is possible that *FM* contains some artifacts, so a morphological opening is applied to remove the smallest ones, followed by a morphological region filling to fill the holes that may be present inside *FM* and a morphological closing is then applied to smooth the contour by filling gaps that may still be present at this stage. The result is shown in Figure 2 (c). Finally, the segmented foreground image is shown in Figure 2 (d).

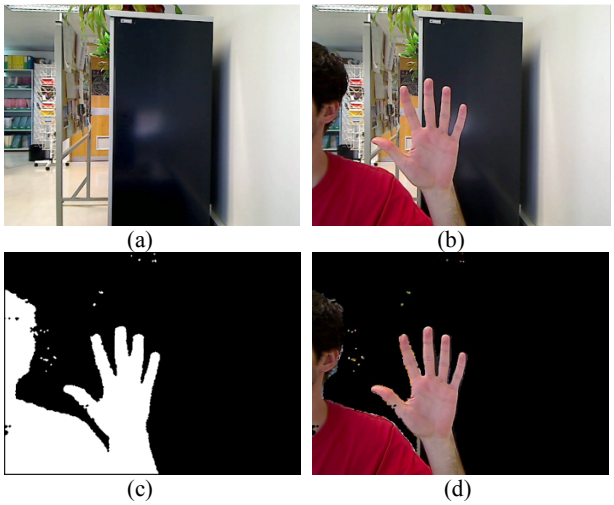


Figure 2 – Background subtraction: (a) – background image; (b) – current frame; (c) – binary mask; (d) – segmented image.

2.2. Skin Color Detection

Since background subtraction segments all moving regions as foreground, a skin color detection technique is applied to segment the candidate hand regions. In this paper, a simple and fast skin detection technique was implemented, explicitly defining the skin color as a region in the RGB color space. A pixel is classified as skin if it satisfies the following conditions [15]:

$$\begin{aligned} &R > 95 \,\&\, G > 40 \,\&\, B > 20 \,\& \\ &\max\{R,G,B\} - \min\{R,G,B\} > 15 \,\& \\ &|R - G| > 15 \,\&\, R > G \,\&\, R > B \end{aligned} \tag{3}$$

In Figure 3 (a), the above-mentioned skin detection technique was applied to the foreground image (Figure 2 (d)). If skin detection had been applied directly to the captured frame, Figure 2 (b), other objects with skin-like color, like a wooden door and some book covers, would also have been detected along with the face, hand and arm (see Figure 3 (b)).

Figure 3 – Skin color detection results in the: (a) foreground image; (b) captured frame.

2.3. Hand Detection and ROI Segmentation

The output of the skin detection module, shown in Figure 3 (a), is binarized (see Figure 4 (a)) and the biggest connected

component is kept (see Figure 4 (b)) and its contour is traced. Then, an automatic verification of whether it corresponds to a hand with stretched fingers is performed, notably by computing the radial distance between all hand contour points and a reference point (Figure 4 (c)), which may lie in the wrist or arm regions [16]; and checking whether five maxima (fingertips) and four minima (finger webs) points are detected.

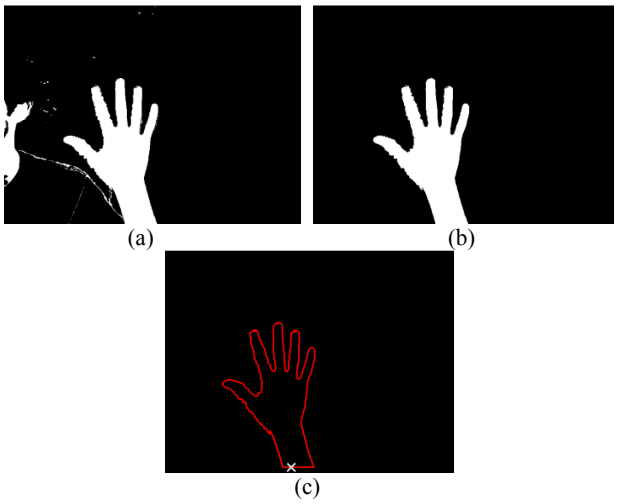


Figure 4 – Candidate hand region detection: (a) – binary mask corresponding to Figure 3 (a); (b) – largest blob detected in (a); (c) – hand contour with reference point;

Additionally, to obtain more accurate locations, a contour curvegram [16] is computed around the previously obtained points and the points with maximum curvature are taken as the final fingertip and finger-web locations.
Two additional reference points are computed for the palm ROI extraction (see Figure 5). The palm region is defined by a square, whose vertices are obtained by computing the midpoint between the additional reference points and the finger-webs of the index and little fingers. The extracted square is rotated to a vertical position and resized to 128×128.

2.4. Background Update

In order to adapt to the changes observed in the acquired images (e.g., change in illumination, objects being introduced or removed from the scene), a background update algorithm is implemented. The first step is to detect if there are moving objects in the current frame by computing the frame difference between two consecutive frames using the method described in Section 2.1. If movement is detected, the background should not be updated. Remember that, in this case, the foreground mask, *FM*, contains the movement detected from the previous to the current frame. In order to avoid false positives in movement detection, e.g., due to noise, a threshold (γ) is

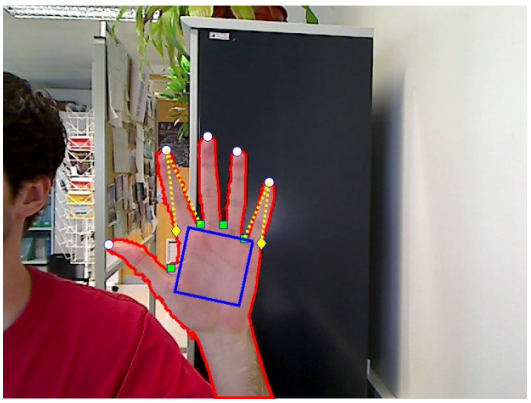


Figure 5 – Hand feature points detection and ROI extraction.

applied.

$$\mu = \left[\sum_{x=1}^{width} \sum_{y=1}^{height} FM(x,y) \right] > \gamma, \tag{4}$$

where γ is a background-dependent threshold, which has to be initialized whenever the camera's position is changed. The updated background is given by

$$B_{i+1} = \begin{cases} B_i, \text{ if } \mu = 1 \text{ or hand detected in } F_i \\ \alpha F_i + (1-\alpha) B_i, \text{ otherwise} \end{cases}, \tag{5}$$

where B_i is the current background, F_i is the current frame and α is the learning rate. In this paper, $\alpha = 0.05$.

2.5. Feature Extraction

In this paper, a state-of-the-art palmprint feature extraction algorithm, OLOF, has been implemented. This technique has been previously used to extract features from palmprint texture [17]. The filters used in this technique are given by

$$OF(\theta) = f(x,y,\theta) - f\left(x,y,\theta + \pi/2\right), \tag{6}$$

$$f(x,y,\theta) = \exp\left[-\left(\frac{(x-\mu_x)\cos\theta + (y-\mu_y)\sin\theta}{\sigma_x} \right)^2 - \left(\frac{-(x-\mu_x)\sin\theta + (y-\mu_y)\cos\theta}{\sigma_y} \right)^2 \right] \tag{7}$$

where θ denotes the differential Gaussian filter's orientation, μ_x, μ_y, σ_x and σ_y are the filter's means and standard deviations in x and y axes, respectively. In this paper, the filter size is 35×35 pixels; $\mu_x = \mu_y = 17$, $\sigma_x = 9$ and $\sigma_y = 3$. For each pixel in the palmprint ROI, the convolution with three differential Gaussian filters, $OF(0)$, $OF(\pi/6)$, $OF(\pi/3)$, is performed. Only the signs of the convolution results are kept in the feature vector.

2.6. Secure Template Storage

The proposed secure template storage module is illustrated in Figure 6. When a user is enrolled five templates are

registered in order to account for some of the intra-user variations. A set of parity bits, $[p_1 \ldots p_5]$, is computed by the LDPC encoder from the user's templates $[b_1 \ldots b_5]$. Parallel to this process, the bitwise exclusive disjunction (XOR) between $[b_1 \ldots b_5]$ and a randomly generated word, w, is computed. This is done for two reasons: (i) templates from the same person are different in distinct biometric systems; (ii) if a template is compromised, a new one can be issued by changing w.

The result, $[x_1 \ldots x_5]$, is processed by a CHF to conceal their values and the output, $[h_1 \ldots h_5]$, is stored in the database. A user noise model, η, is also computed. It results from the comparison of the user's five templates (i.e., a total of 10 comparisons). In each comparison, the value of η_i is updated with the probability of the i-th bit changing its value due to intra-user variations. This will give the decoder a good measure of bit confidence while not revealing any information about the bit value itself. Hence, the user's template consists of (p, w, h, η) and is associated with a unique ID.

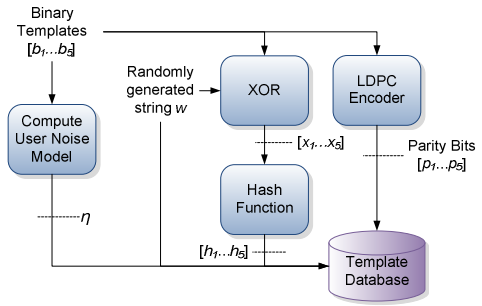


Figure 6 – Secure template storage block diagram.

2.7. Secure Template Matching

The probe binary template, b', is used as input of the secure template matching module (see Figure 7). Since the LDPC decoder receives probabilities instead of bit values, the Log-Likelihood Ratio (LLR) must be computed.

$$LLR(b_i \mid b_i') = \log\left(\frac{P(b_i = 0 \mid b_i')}{P(b_i = 1 \mid b_i')}\right), \quad (8)$$

where $P(b_i = 1 \mid b_i')$ is the probability of the i-th bit in b being 1, given the observed value in b_i'. Since the value in η_i corresponds to the estimated probability of b_i changing value, the LLR is computed with the following values

$$P(b_i = 0 \mid b_i') = \begin{cases} 1 - \eta_i, & \text{if } b_i' = 0 \\ \eta_i, & \text{if } b_i' = 1 \end{cases}, \quad (9)$$

$$P(b_i = 1 \mid b_i') = \begin{cases} \eta_i, & \text{if } b_i' = 0 \\ 1 - \eta_i, & \text{if } b_i' = 1 \end{cases}. \quad (10)$$

The LDPC decoding process is iterative and done by Belief Propagation. In this paper, the maximum number of iterations is 20 since experiments revealed that more iterations degraded the recognition speed while not improving the recognition accuracy. If the decoding is successful, the hash value h' matches the stored hash value, h, and the user is identified. Otherwise, the identification algorithm takes the next ID in the database and repeats the process. If no more IDs are available, the user is not identified.

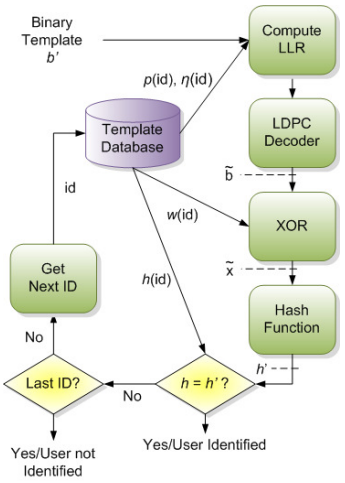


Figure 7 – Secure template matching block diagram.

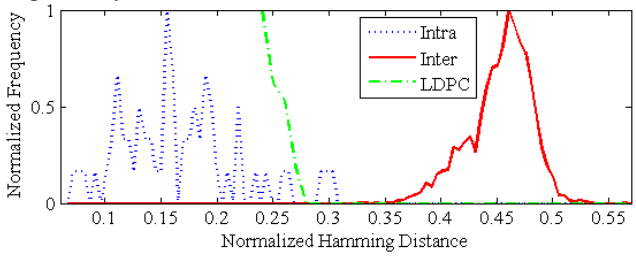
3. EXPERIMENTAL RESULTS

To perform the experiments, a Logitech Webcam Pro 9000 was used together with a MATLAB R2009b implementation. The computer has an Intel® Core™ i7 860 processor and 8 GB of RAM. The webcam resolution was set to 800 by 600 pixels and auto focus, auto white balance, auto gain and auto exposure, were disabled to avoid any undesirable changes in the acquired images.

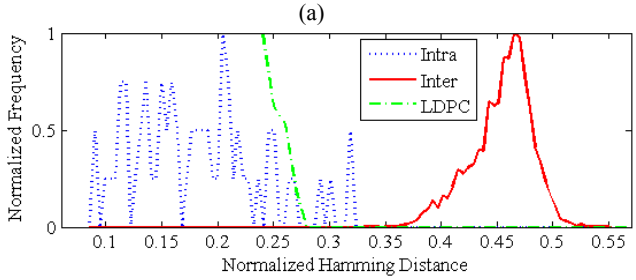
The experiments were conducted on a palmprint database collected using an implementation of the proposed system. All images were acquired in an indoor environment with natural and artificial light, similar to Figure 2 (b). The distance between the hand and the camera ranged from 30 to 65 cm. The database contains a total of 960 images: 15 left and 15 right palmprints from 32 users. All right palmprint images were mirrored so that their orientation is the same as left palmprint images. The objective is to consider left and right palmprint images from the same person as left palmprint images from two different persons, thus increasing the number of subjects in the databases. Therefore, a total of 64 subjects is considered. The database is split into 3 sets: registration, training and test, each containing five images from each hand's palmprint. In the training session, the registration and training sets are compared, using a Hamming distance metric (HD), to determine the maximum bit error rate (BER) that should be corrected by the LDPC code. In this paper, BER is defined as the bit error rate measured at the decoder's input side. Each palmprint in the training set generates 5 intra-user and

285 inter-user comparisons. In the testing session, the same number of comparisons is performed and the previously designed LDPC code is used to evaluate the recognition accuracy, in terms of false acceptance and false rejection rates, FAR and FRR, respectively.

After performing the training session, a (3072,2850) LDPC code is designed to correct templates with a BER of, at most, 28% (see Figure 8 (a)). The parity-check matrix, H, has a fixed number of 3 ones per column and a variable number of ones per row: $\rho_4 = 0.7663$ and $\rho_5 = 0.2337$ represent the ratio of rows that contain 4 and 5 ones, respectively.



(a)



(b)

Figure 8 – Genuine and impostor distributions of: (a) – the training session; (b) – the testing session.

The maximum BER was chosen to be 28% because this value guarantees a 0% FAR. The resulting FRR is 3.17% in the training session and 5% in the test session.

4. CONCLUSIONS

This paper presents an unconstrained palmprint recognition system with secure template storage. The proposed system has been successfully tested in indoor environments with natural and artificial light. Identification performance results show that it is possible to achieve a 0% FAR with a 5% FRR. Some user cooperation is required (e.g., placing the hand within the operating range, stretching the fingers), which makes the system suitable for applications where it is of the user's interest to be identified. The proposed system also has some limitations: the hand detection fails when two skin regions (e.g., the hand and the face) overlap or if the hand is placed too close to the camera.

5. REFERENCES

[1] A K Jain, A Ross, and S Prabhakar, "An Introduction to Biometric Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4-20, January 2004.

[2] W Shu and D Zhang, "Automated Personal Identification by Palmprint," *Optical Engineering*, vol. 37, no. 8, pp. 2359-2362, August 1998.

[3] D Zhang, W-K Kong, J You, and M Wong, "Online Palmprint Identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1041-1050, September 2003.

[4] T Connie, A T B Jin, M G K Ong, and D N C Ling, "An Automated Palmprint Recognition System," *Image and Vision Computing*, vol. 23, no. 5, pp. 501-515, May 2005.

[5] Miguel Ferrer, Aythami Morales, Carlos Travieso, and Jesús Alonso, "Low Cost Multimodal Biometric identification System Based on Hand Geometry, Palm and Finger Print Texture," in *41st Annual IEEE International Carnahan Conference on Security Technology*, 2007, pp. 52-58.

[6] C C Han, H L Cheng, K C Fan, and C L Lin, "Personal Authentication Using Palm-print Features," *Pattern Recognition*, vol. 36, no. 2, pp. 371-381, February 2003.

[7] S Ribaric and I Fratric, "A Biometric Identification System Based on Eigenpalm and Eigenfinger Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1698-1709, November 2005.

[8] A Kumar and D Zhang, "Personal Authentication Using Multiple Palmprint Representation," *Pattern Recognition*, vol. 38, no. 10, pp. 1695-1704, October 2005.

[9] C Methani and A Namboodiri, "Pose Invariant Palmprint Recognition," in *IEEE International Conference on Biometrics*, Alghero, Italy, 2009, pp. 577-586.

[10] Y Han, Z Sun, F Wang, and T Tan, "Palmprint Recognition Under Unconstrained Scenes," in *Proceedings of the 8th Asian Conference on Computer Vision*, Tokyo, Japan, 2007, pp. 1-11.

[11] G.I. Davida, Y. Frankel, and B.J. Matt, "On Enabling Secure Applications Through Off-line Biometric Identification," in *IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 1998, pp. 148-157.

[12] A Juels and M Wattenberg, "A Fuzzy Commitment Scheme," in *Proc. of the Sixth ACM Conference on Computer and Communications Security*, Singapore, 1999, pp. 28-36.

[13] A Juels and M Sudan, "A Fuzzy Vault Scheme," in *Proceedings of IEEE International Symposium on Information Theory*, Lausanne, Switzerland, 2002, p. 408.

[14] A Nagar, S Rane, and A Vetro, "Alignment and Bit Extraction for Secure Fingerprint Biometrics," in *SPIE Conference on Electronic Imaging 2010 (Special Collection)*, San Jose, CA, USA, 2010, pp. Vol. 7541, 75410N (2010).

[15] J Kovac, P Peer, and F Solina, "Human Skin Color Clustering for Face Detection," in *The IEEE Region 8 EUROCON on Computer as a Tool*, Turku, Finland, 2003, pp. 144-148.

[16] E Yörük, E KonukoGlu, B Sankur, and J Darbon, "Shape-Based Hand Recognition," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1803-1815, July 2006.

[17] Z Sun, T Tan, Y Wang, and S Z Li, "Ordinal Palmprint Representation for Personal Identification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, 2005, pp. 279-284.