A Fuzzy System for Three-Factor, Non-textual Authentication

James Stockdale, Alex Vakaloudis, Juan Manuel Escaño, Jian Liang and Brian Cahill

Abstract As text-based authentication has had its critiques, non-textual techniques have been suggested throughout the last two decades. However, it is only lately, with the wide-spread adoption of smartphones and tablet devices that they have found a compelling application. Non-textual authentication may be faster and more secure and it also introduces a new paradigm for the authentication decision. We present a three factor system based on facial recognition, gesture and device ID and we define a fuzzy matching engine to handle authentication. Preliminary results indicate that such an approach can be fast and user-friendly.

Keywords Fuzzy matching · Authentication · Biometric recognition · Gesture recognition · Multi-factor

1 Introduction

Passwords are a familiar, perhaps ubiquitous feature of everyday modern life. The conventional authentication paradigm invokes a username to identify and a password to authenticate an individual user. Typically, these two elements of the

J. Stockdale · A. Vakaloudis (✉) · J.M. Escaño · J. Liang · B. Cahill
Nimbus Centre, Cork Institute of Technology, Bishopstown, Cork, Ireland
e-mail: alex.vakaloudis@cit.ie

J. Stockdale
e-mail: james.stockdale@cit.ie

J.M. Escaño
e-mail: juanmanuel.escano@cit.ie

J. Liang
e-mail: jian.liang@cit.ie

B. Cahill
e-mail: brian.cahill@cit.ie

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_8

authentication process are in a textual form. Much has been written about what constitutes a secure password and increasingly, users are advised, if not required, to provide ever longer and more complex passwords in the interest of maintaining security.

With the modern proliferation of both hardware and software that are designed to be largely operated without the use of a keyboard, the entering of traditional, textual passwords has become something of a chore, an inconvenience and, perhaps, an anachronism. Non-textual methods of authentication have been suggested and, more recently, implemented. Well known examples include gesture recognition implemented on various smartphones and biometric systems such as Apple Inc. Touch ID and facial recognition as supported by Android based mobiles.

Non-textual authentication methods differ in a number of ways from the classical username-password approach. Key among these is that successful authentication follows not only from an exactly matching input, but from any one of the set of sufficiently matching inputs. While the textual password must exactly match the stored prototype, the non-textual input need only be sufficiently similar to the stored prototype since the exact match is exceedingly unlikely. The requirement for a proximity based match suggests that a fuzzy approach is appropriate. In this paper we describe a three-factor authentication system employing fuzzy matching to determine the degree of matching between non-textual elements of authentication data.

2 Related Work

Fuzzy logic [1] has been widely used in matching techniques [2]. Various approaches are employed, such as fuzzy transforms [3, 4], relative distance [5–7] and similarity measure [8, 9]. Typical applications of fuzzy matching are text and signature recognition, due to the ability of characters to convey the same information while taking on different graphical forms [10–14].

Since gestures cannot be repeated with precision, but can convey sufficient information to consider them *almost equal* to a stored prototype, fuzzy logic is a suitable technique to check for similarity. To recognize faces, an extraction of features can be performed using a biometric algorithm. Authentication based on a biometric factor is a widely used technique for mobile devices e.g. [15]. Fuzzy logic is also an established method for matching those features [16].

3 System Design and Development

3.1 System Description

The System developed implements a three-factor authentication service using biometric, gesture and device id as the three factors. In usage, a user is presented with a camera view of himself and required to click the screen to freeze the image. The

user then draws a simple image, the gesture, on top of the frozen image. Example gestures may be a smile, a hat, a moustache, spectacles, etc., or perhaps something more abstract. If the user is registering a new account, this procedure must be repeated a certain number of times so that the system can confirm that the biometrics and gestures are sufficiently similar. This is analogous to the *repeat password* prompt that is familiar in textual authentication systems. In addition, the system tests the new user's biometric for absence of similarity to all previously registered biometrics and returns an error is a similar biometric is found. This is analogous to a *user id already in use* message in traditional systems. If the user has already registered and is returning to login, the process is performed once and authentication (or not) is determined based on the captured biometric and gesture. In practice, each user account is also tied to a specific device. The device is determined during the first registration and subsequent logins may only be authenticated for that user when using the same device. Therefore, the three factors of authentication in our system are biometric (who I am), gesture (what I know) and device (what I have).

3.2 Fuzzy Matching Engine

Since 1975, many engineering applications have been developed based on the use of fuzzy logic [17]. Fuzzy systems handle information closer to the human way, i.e., uncertain, vague or imprecise. In the model proposed by Takagi–Sugeno (TS) [18], the structure of antecedent describes fuzzy regions in the input space, and that of consequent presents non-fuzzy functions of the model inputs. The system may be described for each rule as follows:

R_j:
IF $x_1(k)$ is $F_{1j}, \ldots,$ and $x_n(k)$ is F_{nj},
THEN:
$y_j(k) = y_j$

where y_j is a constant value $X(k) = [x_1(k)x_2(k), \ldots, x_n(k)]^T$ is the input vector of the fuzzy system in the instant k, F_{ij} is the fuzzy set respective to $x_i(k)$ on the rule j, $y_j(k)$ is the output of the model respect to the operating region associated with the rule. If $\mu_{ij}(k)$ is the membership degree of $x_i(k)$ in the fuzzy set F_{ij} and the number of implications or rules is L, the complete model can be described by

$$y(k) = \sum_{j=1}^{L} w_j(k) y_j \tag{1}$$

where

$$w_j(k) = \frac{\bar{\mu}_j(k)}{\sum_{j=1}^{L} \bar{\mu}_j(k)}, \quad \bar{\mu}_j(k) = \prod_{i=1}^{n} \mu_{ij}(k)$$

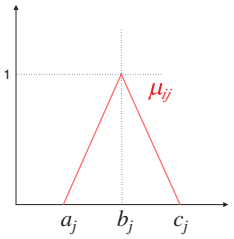

Fig. 1 Triangular function

In this application we have used triangular membership functions, defined as:

$$\mu_{ij}(k) = \begin{cases} 0 & x < a_j \\ \frac{x_i(k)-a_j}{b_j-a_j} & a_j \leq x \leq b_j \\ \frac{c_j-x_i(k)}{c_j-b_j} & b_j < x < c_j \\ 0 & x \geq c_j \end{cases} \tag{2}$$

or in a compact form,

$$\mu_{ij}(k) = \max\left[\min\left(\frac{x_i(k)-a_j}{b_j-a_j}, \frac{c_j-x_i(k)}{c_j-b_j}\right), 0\right] \tag{3}$$

where a_j, b_j and c_j are parameters which define the triangular function, as shown in Fig. 1.

A position is composed of two numbers. The prototype will be composed of N points with the positions

$$P = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$$

A first prototype to be registered is built by the average of the position of several gestures given by the registration process. The FME will make an index using the degree of membership of each pattern point to the prototype point.

There will be a fuzzy number defined for each prototype coordinate:

$$\tilde{P} = \{(\tilde{x}_1, \tilde{y}_1), (\tilde{x}_2, \tilde{y}_2), \ldots, (\tilde{x}_N, \tilde{y}_N)\}$$

The fuzzy number will be defined for the couple $\{b, d\}$ where b is the representative crisp number of $\tilde{b}$ and d will be an adjusting parameter which defines the distance $c - a$. In order to simplify the application, we will set it up with the same value for all the fuzzy numbers, calling it the *fuzziness* parameter.

Using a rule like: IF x_i IS $\tilde{x}_i$ THEN $y = 1$, the degree of membership $\mu_{\tilde{x}_i}(x_i)$ of the crisp number x_i to the fuzzy number $\tilde{x}_i$ is obtained. Applying the rule to each coordinate gives a set of $\{\mu_{\tilde{x}_i}(x_i),\ \mu_{\tilde{y}_i}(y_i)\}$. Taking into account the sequence order and calculating each degree of membership, the expression

$$\mu = \frac{\sum_{i=1}^{N} \mu_{\tilde{x}_i}(x_i) \cdot \mu_{\tilde{y}_i}(y_i)}{N} \tag{4}$$

yields a matching index for the gesture and feature vector. A threshold value can then be used to establish whether the index value represents a match or not. This parameter is referred to as the *sensitivity*.

3.3 Simulation Result Using the FME

Figures 2 and 3 show examples of matching (after adjusting fuzziness and sensitivity) using the matching index (4).

Before applying the FME, the gesture is normalised to an image of the user's face in terms of orientation and scale so that comparisons can be made. For instance, the vector formed by joining the center of the eyes is a good reference. Figure 4 show how the gesture is matched with different orientations and sizes.

3.4 Computation of a Biometric

There are many types of biometric indicators that could be used within a non-textual authentication system. For example, fingerprint, palm print, iris, DNA, etc. However, facial recognition is a desirable choice because it requires only a camera, which is now a fairly ubiquitous component of modern mobile, laptop and desktop devices. Therefore, facial recognition is a suitable component for a system that will be rolled out across a wide range of modern devices.

Since the FME prefers to work with a vector of numerical values, a simple biometric that distils a face down to five numbers was chosen. Using OpenCV's [19] object detection library and, specifically, cascade classifiers, four prominent features of the face are detected. These are namely, the nose, the mouth and the two eyes. Each is defined by the rectangular region that encloses it. By computing the distance between the centre of each of these rectangles, a vector of six values is obtained. By assuming one of these distances to be of unit length and normalising the other values against it, a biometric descriptor comprising five meaningful values remains.

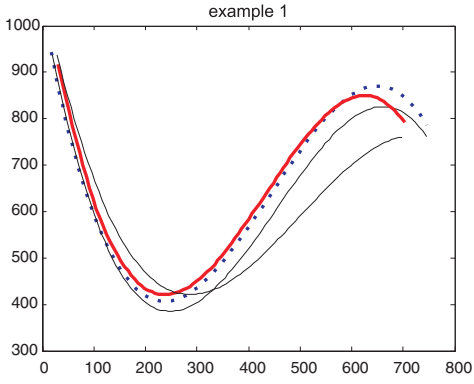

Fig. 2 Example of matching. *Dot line* prototype; *Thick solid line* matched gesture

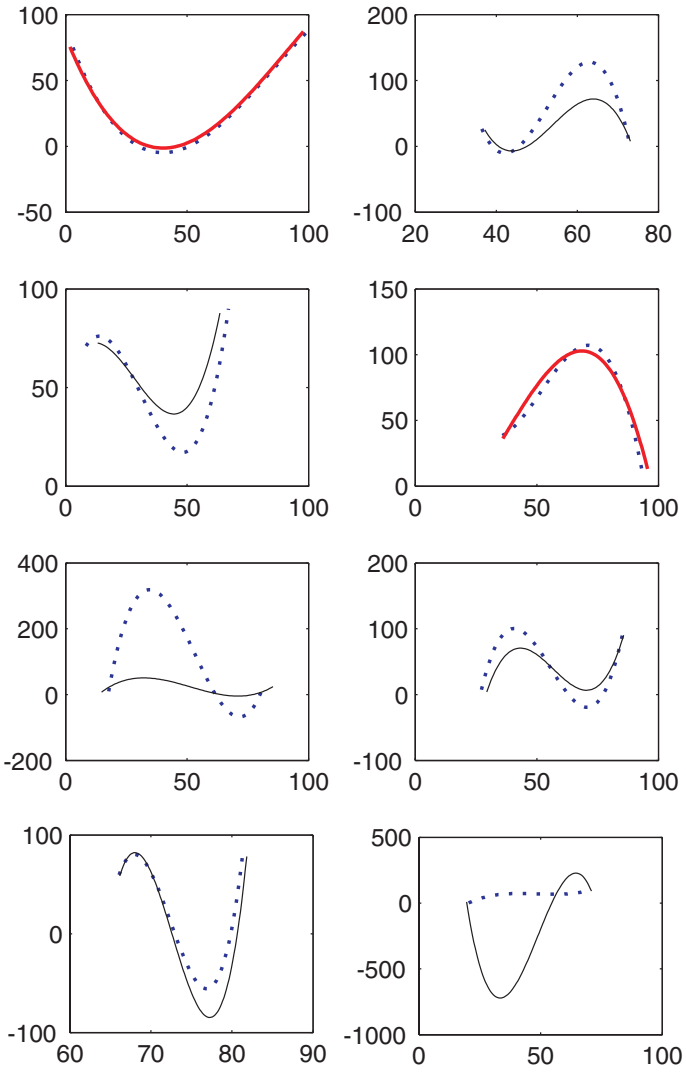

Fig. 3 Examples of matching. *Dot line* prototype; *Thick solid line* matched gesture

3.5 Gesture Capture

A gesture can be described as a sequence of coordinate pairs and is captured while the user completes a drag operation using an available pointing device. On a mobile or tablet device, this will normally be accomplished by touching and dragging on the screen while a desktop computer user may use a mouse or a trackball. On a laptop, perhaps all of these options may be available. In any case, after the

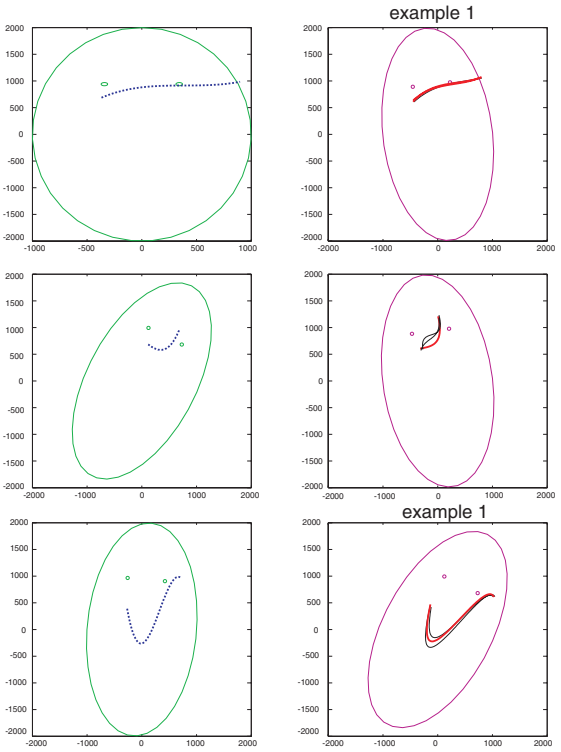

Fig. 4 Example of matching after normalisation. *Dot line* prototype; *Thick solid line* matched gesture

drag operation is completed, a sequence of coordinate pairs will have been captured. In their raw form, these normally represent absolute pixel positions on the device and there may be very many or very few pairs depending upon whether the gesture was drawn slowly or quickly.

To make the gestures more easily comparable, they are first standardised. This simply involved adding extra points or removing extraneous points in order to achieve some predetermined number of coordinate pairs. It is important that the process of standardisation does not materially alter the overall shape of the gesture. It is possible to use the FME to compare a gesture pattern to a stored prototype as long as both have been standardised to the same number of points.

The final step to ensure that gestures can be compared in a meaningful and repeatable manner is to normalise them to the biometric. In this system, a gesture will always be associated with a biometric descriptor. Normalisation takes the gesture out of the device specific, pixel based coordinate system that it originates in and converts it to a space that is determined by the size, location and orientation of the biometric. For this purpose, the vector joining the centre of the eyes is used. This vector defines the unit length along the x-axis in the normalised coordinate space. This allows for the natural variations that will result from users presenting themselves to the camera inconsistently. Perhaps sometimes to one side or to the other, perhaps sometimes closer or farther away. The result is that the image of the

face—the part of the image that generates the biometric will often appear in different parts of the overall camera frame and may take up differing proportions of it. Normalisation eliminates these differences, essentially ensuring that all biometric-gesture pairs are meaningfully comparable. In simple terms, if you have a bigger head, you need to draw a bigger hat.

3.6 Parameterisation

It has been stated that the FME is controlled by simply two parameters, namely fuzziness and sensitivity. The former is applied to the individual differences between elements within a prototype-pattern pair while the latter applies to the aggregation of the scores determined from these differences. However, within this system, it is clear that the FME is used in a number of different contexts and that different fuzziness-sensitivity parameter pairs may be needed for some of these various contexts.

Broadly speaking, the FME is used in two main roles, namely biometric recognition and gesture recognition. However, these roles are performed in two distinct life-cycle phases of the system, namely registration and authentication. Arguably, the system may be more or less lenient depending on the life-cycle phase, thus requiring different parameter pairs for the two roles. Furthermore, as has been previously stated, during registration, the biometric is checked for similarity with other biometrics in the registration process. However, prior to this it is checked for uniqueness against other stored biometrics in the database. This introduces yet another context, which is distinct from all of the others in that it tests for uniqueness (or, more correctly, absence of similarity) as opposed to similarity.

Thus, there are five distinct contexts in which the FME is used and for which an independent fuzziness-sensitivity parameter pair can be defined. Table 1 enumerates the ten possible parameter values and also shows the name ascribed to each parameter within the system. It can be seen from this table that each of the parameters is not independently variable within our system. Rather, the fuzziness for biometric matching, F_b is repeated across *all* biometric matching contexts.

Table 1 Parameters for controlling behaviour of the fuzzy matching engine

	Phase	Object	Comparison	Parameter	Name
1	Reg	Biometric	Similarity	Fuzziness	F_b
2	Reg	Biometric	Similarity	Sensitivity	S_1
3	Reg	Gesture	Similarity	Fuzziness	F_g
4	Reg	Gesture	Similarity	Sensitivity	S_2
5	Reg	Biometric	Uniqueness	Fuzziness	F_b
6	Reg	Biometric	Uniqueness	Sensitivity	S_3
7	Auth	Biometric	Similarity	Fuzziness	F_b
8	Auth	Biometric	Similarity	Sensitivity	S_4
9	Auth	Gesture	Similarity	Fuzziness	F_g
10	Auth	Gesture	Similarity	Sensitivity	S_5

Similarly, F_g, the fuzziness for gesture matching is a single value used in all gesture matching contexts. In contrast, the sensitivities for the five FME contexts are independently variable. Therefore, our system uses a total of seven parameters to control fuzzy matching; two independent fuzziness parameters (F_b, F_g) and five independent sensitivity parameters (S_{1-5}).

3.7 Gradual Migration of Prototype

By the very nature of the system, neither a biometric descriptor, nor a gesture will ever be an identical image of the stored prototype. It is expected that both kinds of pattern will differ from their prototype at all authentication attempts. However, an additional feature of our system allows for the gradual migration the prototype itself in response to the successfully authenticated patterns. A moving window retaining the previous n successfully logged in biometric descriptors and gestures is maintained. At each successful authentication, the newest pair of patterns is added to this window and the oldest is removed. From the window, a mean pattern is computed for both biometric and gesture and this becomes the new prototype that will be compared against during the next authentication attempt. This caters for the scenario that a user may register with a very carefully drawn gesture but that over time, as they become accustomed to using the system, they may adopt a more casual approach to repeating the gesture. However, the system still retains the original prototype that was registered and it is possible to raise an alarm if a user's biometric or gesture has *crept* too far from its original representation. Although we implement this functionality for both biometric and gesture, in practice, we expect that it is really only useful in the latter context.

3.8 Forgotten Gestures

Just as the user of a traditional textual authentication system may forget their secret password, it may occur that a user of our system forgets, or is unable to satisfactorily repeat their registered gesture. In this case, we offer the facility to reset the gesture component of the user's login credentials. When this happens, the user is invited to provide a new gesture, which is analogous to providing a new password within a textual system. The process is similar to registration in that the gesture must be repeated three times on top of three different images. However, in this mode, the biometric is not stored as a prototype but rather compared for similarity against the existing stored prototype. Similarly in this mode, the device id is also checked to confirm that it is the correct device. Therefore, only one of the three factors is reset while the other two serve as authentication during this process. Additional security can be provided by, for example, ensuring that the gesture reset must be performed within a certain duration after the reset is issued.

In principle, it would be possible to apply this strategy to any of the three factors. A user may wish to move their account to a different device. In this case, a device reset could be issued, allowing the user to authenticate themselves using only biometric and gesture on a new device. Similarly, but perhaps less realistically, the biometric factor could be reset, allowing the user to register a new biometric while authenticating themselves by gesture and device id.

4 Implementation and Results

The case studies for applying this research work are diverse, ranging from gaining local access on a native application to authenticating against cloud systems. We also need to consider usability and security constraints, for example whether facial detection takes place remotely or locally. Consequently, there are a number of different architectures that can be implemented around the core of the fuzzy matching engine. To cover as many cases as possible we have implemented two separate architectures discussed below.

The first approach (Fig. 5) is a cloud based architecture. We used an HTML5 client to capture the face and the gesture. The WebRTC [20] standard enables us to take a photo either by a single touch event or when a smile is detected. The user then draws their gesture, which is handled by mouse motion or touch events. Both the bitmap of the face and the array of co-ordinates for the gesture are then sent to the server with REST calls. The server uses the OpenCV library to extract the biometric data which, along with the gesture are passed to the fuzzy engine for authentication. The server is implemented using Java with the Spring Framework and runs on a Glassfish 4 Server, while data are stored in a CouchDB database.

The advantages of this architecture are firstly security, since authentication and data storage are performed remotely and secondly, flexibility deriving from a cloud-like deployment. On the other hand, it is dependent on network connectivity. The client-server architecture means that many unsuitable pictures may be sent to the server before a face is detected. This can means that until the user is familiar

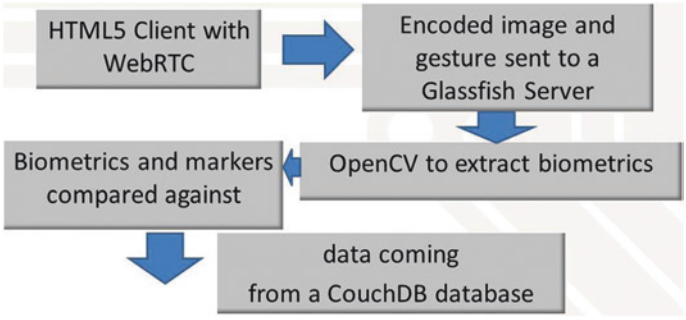

Fig. 5 Cloud implementation

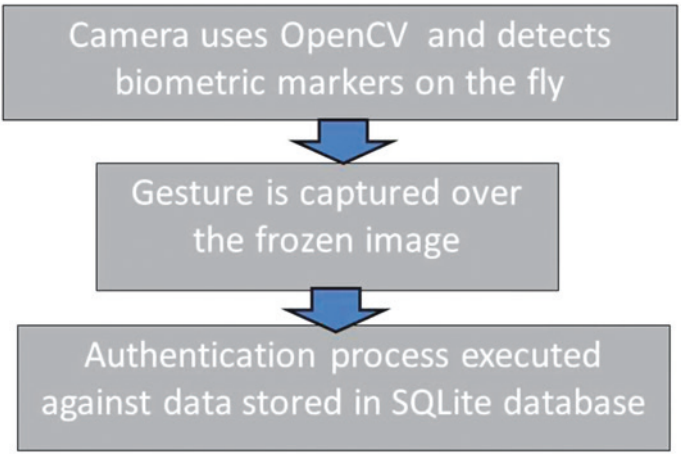

Fig. 6 Android implementation

with the conditions under which successful recognition are likely, using the system can be slow.

The second implementation (Fig. 6) concerns a native application currently for Android devices. OpenCV is once again employed however, the biometric extraction happens in real time as every frame in the video stream is assessed. Preliminary results show that after a short period of self-training, the time required for a user to achieve a sufficient picture is much less than one second. The picture of the face is then frozen for the gesture to take place and similarly to the previous case, data are passed to the fuzzy engine. In this implementation, a SQlite database is used. The strong points of this method are speed and non-reliance on network connectivity. However, having the authentication data locally may be a security vulnerability.

We chose to work on these cases in order to produce a set of modules that could be used in a hybrid implementation in the future. For example, we could use the OpenCV on a native application which sends data to a remote server.

5 Case Study

A pilot study has been conducted for the HTML5 implementation having the main focus to engage with a cohort of people from age 18 to 65 and to observe their interaction with the user experience. All participants were furnished with a basic list of instructions and asked to complete a short online survey.

The pilot accommodated participants with a variety of devices to engage with the system, namely a laptop with external webcam, a tablet and a smart phone. During the course of this 3 day pilot, 19 participants successfully registered on the

system. Of these, 17 participants experienced at least one successful login while 2 participants failed to login. In observing the participants, problems arose when using the laptop and webcam, due primarily to the positioning of the webcam and the laptop touchpad. Eye contact with the system is limited in this scenario. Best results were observed when using a tablet and smart phones. Nine participants successfully completed the online survey, five participants did not complete the survey while five participants failed to engage with the survey at all.

The most obvious of these finding was the necessity to use a tablet or a smart phone. A laptop with external webcam connected will work but will require additional patience and attention to detail from the end user. Most participants agreed that their experience was a positive one. Most agreed that they could use such an access system when using their laptops, tablets, kindle, and smart phones. They were not so confident in using the system when under time constraints or in a scenario when others are waiting to access the same device (an ATM for example).

6 Conclusion

We presented a system for multi-factor authentication based on a fuzzy matching engine. We applied fuzzy matching in two factors namely, biometric (facial recognition) and knowledge (gesture). Non-textual authentication differs from the traditional username-password approach; there is no unique matching and moreover there is a weak dependency between the biometric and the knowledge part. We exploited the latter one by normalising the gesture over the face.

We also defined the parameters that influence the security of this fuzzy-based approach and we outlined its implementation both as a cloud-based or native application. While possible areas of application are limitless, for the foreseeable future we consider e-learning and people with special needs.

Future work involves developing a training system for automatically defining values for fuzziness and sensitivity given certain security constraints, the use of other biometric techniques (e.g. fingerprints) and incorporation within the core of operating systems.

Acknowledgments This work was supported by Enterprise Ireland and carried out under the intellectual property of Sensipass Ltd. Patent Publication No. WO/2012/164385 Method and Computer Program for Providing Authentication to Control Access to a Computer System, Roman Sirota (UA), Michael J. Hill (US) and Thomas R. Ruddy (US).

References

1. Ross, T.J.: Fuzzy Logic with Engineering Applications, Wiley, New York (2004). ISBN: 0470860758. http://www.worldcat.org/isbn/0470860758
2. Chi, Z., Yan, H., Pham, T.: Fuzzy algorithms: with applications to image processing and pattern recognition. In: Advances in Fuzzy Systems—Applications and Theory, vol. 10. World Scientific (1996). ISBN: 9810226977, 9789810226978

3. Martino, F.D., Sessa, S.: Image matching by using fuzzy transforms. Adv. Fuzzy Syst. **2013**(760704), 10 (2013). doi:10.1155/2013/760704
4. Perfilieva, I.: Fuzzy Transforms, Transactions on Rough Sets II, vol. 3135, pp. 63–81. Lecture Notes in Computer Science. Springer (2005). ISBN: 978-3-540-23990-1
5. Bloch, I: Fuzzy relative position between objects in image processing: a morphological approach. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (1999)
6. Bloch, I., Ralescu, A.: Directional relative position between objects in image processing: a comparison between fuzzy approaches. Pattern Recogn. **36**, 1563–1582 (2003)
7. Tan, Q., Akimoto, M.: Fuzzy matching for robot localization. In: Proceedings of IROS. IEEE (1996). ISBN: 96. 0-7803-3213-X
8. Jinwen, T., Jianzhong, H., Jian, L., Dchua, L.: Image matching based on fuzzy information. In: 3rd International Conference on Signal Processing 1996, vol. 2, pp. 946–949. 14–18 Oct 1996. doi:10.1109/ICSIGP.1996.566246
9. Wu, H., Chen, Q., Yachida, M.: Face detection from color images using a fuzzy pattern matching method. IEEE Trans. Pattern Anal. Mach. Intell. **21**(6), 557–563 (1999)
10. Surajit C., Kris, G., Venkatesh, G., Rajeev, M.: Robust and efficient fuzzy match for online data cleaning. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD'03) (2003)
11. Zvi G., Alberto, A.: Pattern Matching Algorithms. Oxford University Press, Oxford (1997). ISBN: 0-19-511367-5
12. Mustafa, A.A.Y.: Fuzzy shape matching with boundary signatures. Pattern Recogn. Lett. **23**, 14731482 (2002)
13. Ukkonen, E.: Algorithms for approximate string matching. Inf. Control **64**, 10018 (1985). doi:10.1016/S0019-9958(85)80046-2
14. Li, Z.K., Xu, L.J., Fang, J., Peng, Q.J., Wang, M.: Research on the surrounding traffic flow of railway station based on License Plate Recognition and fuzzy matching. IEEE (2011). 978-1-61284-109-0
15. Schultz P.T., Sartini, R.A.: Multi factor authentication method and system for multi-factor biometric authentication. US 20130227651 A1 (2012)
16. Vyas, R., Garg, G.: Face recognition using feature extraction and neuro-fuzzy techniques. Int. J. Electron. Comput. Sci. Eng. (2013). ISSN: 2277–1956
17. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
18. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst., Man Cybern. **15**(1), 116132 (1985). http://www.hi.cs.meiji.ac.jp/takagi/paper/TS-MODEL.tar.gz
19. http://www.openCV.org/
20. http://www.webrtc.org/
21. http://argodata.com/solutions/analytics/fuzzy-search/
22. http://dev.w3.org/html5/html-author/
23. Huang, X et al.: A generic framework for three-factor authentication: preserving security and privacy in distributed systems. IEEE Trans. Parallel Distrib. Syst. **22**(8), 1390–1397 (2011)
24. Pulli, K., Baksheev, A., Kornyakov, K., Eruhimov, V.: Real-time computer vision with OpenCV. Commun. ACM (CACM) **55**(6), 61–69 (2012)