TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 19, No. 5, October 2021, pp. 1517~1528
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i5.19824

Rule-based lip-syncing algorithm for virtual character in voice chatbot

Felicia Priscilla Lovely, Arya Wicaksana
Department of Informatics, Universitas Multimedia Nusantara, Tanggerang, Indonesia

Article Info

Article history:

Received Jan 28, 2021
Revised Jul 14, 2021
Accepted Jul 27, 2021

Keywords:

Audio-driven articulation
HMSAM
Manual mapping
Rule-based lip-syncing
Virtual character

ABSTRACT

Virtual characters changed the way we interact with computers. The underlying key for a believable virtual character is accurate synchronization between the visual (lip movements) and the audio (speech) in real-time. This work develops a 3D model for the virtual character and implements the rule-based lip-syncing algorithm for the virtual character's lip movements. We use the Jacob voice chatbot as the platform for the design and implementation of the virtual character. Thus, audio-driven articulation and manual mapping methods are considered suitable for real-time applications such as Jacob. We evaluate the proposed virtual character using hedonic motivation system adoption model (HMSAM) with 70 users. The HMSAM results for the behavioral intention to use is 91.74%, and the immersion is 72.95%. The average score for all aspects of the HMSAM is 85.50%. The rule-based lip-syncing algorithm accurately synchronizes the lip movements with the Jacob voice chatbot's speech in real-time.

Corresponding Author:

Arya Wicaksana
Department of Informatics
Universitas Multimedia Nusantara
Scientia Boulevard, Gading Serpong, Tangerang-15810, Banten, Indonesia
Email: arya.wicaksana@umn.ac.id

1. INTRODUCTION

Virtual characters appear in broad applications with speech features in real-time e.g., avatars in a video call application, in the movie, and the gaming environment [1]. The speech part requires attention to detail in order to enforce the plausibility of the virtual characters. The underlying key to this is to synchronize the lip movements with the speech accurately in real-time.

In previous works, we have developed a voice chatbot application called Jacob [2]-[4]. Jacob functions as a digital assistant to provide users with pre-registered information. It has a speech-to-text module to process the user input and a text-to-speech module to generate the output speech. It also features a face recognition module called Vision [3] and an artificial intelligence module called the Cleveree [4] to paraphrase answers and summarize conversations. However, Jacob does not have any virtual characters for its conversational agent yet, which is a drawback to the user experience when using Jacob.

In this work, we propose to develop and implement a virtual character for the Jacob voice chatbot. Our goal is to provide the user with a virtual character by accurately synchronizing the virtual character's lip movements with the speech in real-time. The virtual character is integrated and tailored to Jacob voice chatbot. We briefly describe the design process of the virtual character for Jacob, including developing the 3D model. We use the rule-based lip-syncing algorithm in [1] for the synchronization part between the speech and virtual character. We also consider the real-time constraint in implementing the algorithm for Jacob.

The virtual character development uses C# and Unity and integrates the virtual character with the Jacob voice chatbot. We use audio-driven articulation for the acoustic analysis and manual mapping for the algorithm's visual mapping part. These techniques are suitable for real-time applications. We evaluate the virtual character using the hedonic motivation system adoption model (HMSAM) [5]-[7]. The HMSAM is used to measure users' behavioral intention to use and immersion when using Jacob with the added virtual character. Hedonic-motivation systems (HMS) are used primarily to fulfill users' intrinsic motivations.

The rest of this paper is organized as follows. Section 2 briefly describes the related works related to the algorithm. Section 3 describes the preliminaries of the paper. Section 4 explains the research methods. Section 5 presents the experimental results, including the end user computing satisfaction (EUCS). Finally, Section 6 concludes this paper with some discussions on future work.

2. RELATED WORKS

Cassell *et al.* [8] proposed the rule-based generation of facial expression, gesture, and spoken intonation for multiple conversational agents. They developed the framework for the rule-based generation and several generators: gesture, gaze, and facial expression (FACS), to achieve the goal. The system uses Pat-Net to synchronize gestures and gaze with the dialogue at the phoneme level. The dialogue itself is generated automatically using a program and a database of facts. Overall, the system produces an adequate approach toward the rule-based generation of facial expression, gesture, and spoken intonation. However, implementation details are limited, and the practicality of the real-time application approach is not stated, including the performance evaluation of the system.

Poggi *et al.* [9] published work on a conversational agent named Greta, an embodied conversational agent (ECA), generally a believable agent. The definition of a believable agent is one that is able to express emotion and to exhibit a given personality. Greta focuses on believable behavior while interacting with the user. Greta's main characteristic is in its ability to dialog with the user in any application domain whose knowledge has been represented with the appropriate formalism. The formalism required makes it not straightforward to use and integrate with existing voice chatbot applications such as Jacob.

The new method for natural mapping speech to the lip shape was proposed by Zoric and Pandzic [10] in 2005 that uses mel-frecuency cepstrum (MFC) and viseme classes. Instead of focusing on body gesture and facial expression like in the previous two works, Zoric and Pandzic focus more on the face animation of a speaking avatar so that it realistically pronounces the text based on the process called lip synchronization. Lip synchronization is the determination of the motion of the mouth and tongue during speech. The process consists of two main parts: audio to visual mapping and calculating visemes for facial animation. The proposed system uses a phoneme database for fine-tuning the face animation. The dataset in the database is used for training the neural network.

The same approach is also proposed by Llorach *et al.* [11] that uses the lip-sync features for virtual character development on Eliza's web. ECAs as desktop applications present drawbacks for both developers and users; the former must develop for each device and operating system, and the latter must install additional software, limiting their widespread use. Llorach claimed that there were no WebGL-based ECAs with 3D virtual characters with advanced features like Eliza in 2017. Compared with other lip-sync systems, the solution presented in [1] is fast and straightforward to implement. A high level of expertise is not required, and there is no need to use a corpus to train the system. The lip-sync can be applied to several different characters with little effort.

Different approach is done by Suwajanakorn [12] that uses machine learning with training videos to produce lip-sync features. However, this method is not suitable for Jacob's need for ECA due to the machine learning model's long and heavy training process, with many videos as the dataset required that is not available in this work. The training process took three hundred Obama videos with a total of seventeen hours. The study's primary goal in [12] is to generate a video of Obama from his voice and stock footage. This approach yields photorealistic results but requires a large corpus of existing video data of a single person. This method is intensely dependent on the speaker [13] and can be challenging to apply.

3. PRELIMINARIES

3.1. Rule-based lip-syncing algorithm

The workflow of the rule-based lip-syncing algorithm introduced in [1] is shown in Figure 1. The algorithm takes speech as input. The speech in this work is captured from the Jacob voice chatbot in real-time. The speech input is then processed in the acoustic analysis step. The rules are embodied in the visual mapping step. These rules determine the movement of the lip animation for the virtual character.

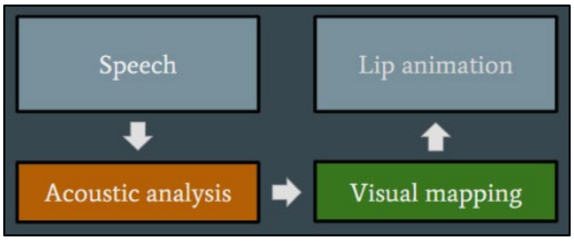


Figure 1. Rule-based lip-syncing workflow

3.1.1. Acoustic analysis

We use a simple energy-based vocal tract model. Formant energy is estimated to produce visual features. The formant is the sound spectral peak, which is thoroughly influenced by the vocal tract [14]. Smoothed short-term power spectrum density (st-PSD) is extracted in real-time by processing each audio block by windowing the input sample using the Blackman window and calculating the fast fourier transform (FFT) [1].

The window technique is used to design the finite impulse response (FIR) filter, a filter of digital signals, which has several types, such as Hanning window, Hamming window, Blackman window, and rectangular window. Blackman window has an advantage in performance and shows the best functionality based on testing in the research [15]. The st-PSD output is smoothed over time with the previous output and then converted to dB. We use (1) for the smoothing.

$$\hat{X}[k] = \tau\, \hat{X}_{-1}[k] + (1-\tau)|X[k]|, for \begin{cases} k = 0, \dots, N-1 \\ 0 < \tau < 1 \end{cases} \tag{1}$$

In (1), $X[k]$ is the complex frequency domain, $\hat{X}[k]$ is the smooth spectrum, and τ is the self-defined smoothing variable, which has a value between 0 to 1. Then, the result is converted to dB, which is denoted as $Y[k]$, producing output, as shown in (2). The results of $Y[k]$, which range from -25dB to -160dB, are then scaled and increased according to the sensitivity threshold symbolized by δ (sensitivity threshold is 0.5 by default). It is then used in (3) to map dynamic range to an interval [-0.5, 0.5], resulting in smoothed st-PSD as $\dot{Y}[k]$ as defined by Llorach [1].

$$Y[k] = 20\, \hat{X}[k]\, , for\ k = 0, \dots, N-1 \tag{2}$$

$$\dot{Y}[k] = \delta + (Y[k] + 20)/140,\ for\ k = 0, \dots, N-1 \tag{3}$$

The results from (3) are divided into frequency bands, and energy is calculated for each frequency band. Energy is calculated using log-scaled data. These frequency bands are defined empirically, and the frequency bins (F_{bins}) are scaled using vocal tract length factor (γ), which is defined by Llorach [1] in (4).

$$F_{bins} = [0, 500\gamma, 700\gamma, 3000\gamma, 6000\gamma]\ Hz \tag{4}$$

The frequency bins are then transformed into frequency data indices (F_{ind}), as an index of each sample in an audio block, as in (5), where fs is the sampling frequency, M is the number of frequency bins and N is the number of samples per audio block [1]. Then, the energy of each bin, which is denoted as $E[m]$, is calculated by only calculating the positive value of st-PSD, which is processed using (6) [1].

$$F_{ind}[m] = \frac{2N}{fs} F_{bins}[m], for\ m = 0, \dots, M-1 \tag{5}$$

$$E[m] = \frac{1}{F_{ind}[m+1] - F_{ind}[m]}\, \Sigma_{j=F_{ind}[m]}^{F_{ind}[m+1]}\, \dot{Y}[j], for \begin{cases} \dot{Y}[j] > 0 \\ m = 0, \dots, M-2 \end{cases} \tag{6}$$

The essential part of this algorithm is the definition of energy bins and limiting frequencies, where the focus is the energy and frequency of the formants of each vowel. References for an average of vowel formants are in Table 1 [16].

Table 1. Average of vowel formants in Hz

Formant	Gender	a	i	u	e	o
F_1	Male	768	342	469	476	497
F_1	Female	936	437	519	536	555
F_2	Male	1,333	2,322	1,122	2,089	910
F_2	Female	1,551	2,761	1,225	2,530	1,035

3.1.2. Visual mapping

After the speech is analyzed, the extracted audio or phoneme feature is mapped to the virtual character's lip parameters, using a viseme, action units, code words, and control parameters. Viseme is a visual phoneme, which is the shape of the lips when speaking a phoneme. This visual parameter determines face deformation and can have different levels of complexity.

There are two types of methods for visual mappings: data-driven mapping and manual mapping. When using several visual parameters, manually mapping audio or phoneme features to these parameters is complicated. Data-driven audio-visual mapping is preferred in the latest research [13], [17], for both audio-driven articulation and phoneme-driven. The data-driven technique requires a corpus that has been recorded to extract the visual parameters related to the acoustic features of phonemes. The results yield high accuracy but highly dependent on the corpus since it is the primary source for creating the animation and the lip trajectory [18].

Most approaches use manual mapping to associate phonemes with specific blend shapes, to handle a small set of comprehensive acoustic and visual parameters. This approach does not require the use of a corpus. A work done by Xu [19] uses 441 animations that are designed manually for the phonemes, and another work by Wang [20] uses 16 blend shapes to map phonemes. The manual mapping is speaker-independent by linking specific frequency features with the visual parameters to produce the animations.

The rule-based lip-syncing algorithm defines a set of frequency-based rules to move three blend shapes to suit different lip configurations, namely kiss, mouth open, and lips pressed. These three blend shapes control each horizontal mouth opening, vertical opening and lip volume, and the various lip configurations that can be achieved. Mouth open is the most basic visual, all vowels, like /a, /i/, /u, /e/, and /o/ using this blend shape. Blend shape model kiss is used to distinguish between vowels, /o/ and /u/ is a representation of this blend shape. For fricative consonants, the blend shape model that is used is when the lips are pressed against each other. This set of functions rules the movement of the blend shape using the values from (7), (8), and (9) [1].

Three visual parameters are calculated using (7), (8), and (9) where $E[m]$ is the energy for bin m that is described in (6) earlier. The BS_{kiss}, BS_{lips}, BS_{mouth} are the weight for each kiss, lips closed, and mouth open blend shape [1]. The coefficients for $E[m]$ in (7), (8), and (9) are obtained from a heuristic approach. The coefficients could be modified accordingly to obtain a different shape of the lips.

$$BS_{kiss} = \begin{cases} 1 - 2E[2], for\ E[1] \geq 0.2 \\ (1 - 2E[2])\ .5E[1], for\ E[1] < 0.2 \end{cases} \tag{7}$$

$$BS_{lips} = \ 3E[3] \tag{8}$$

$$BS_{mouth} = \ 0.8(E[1] - E[3]) \tag{9}$$

The weight of each blend shape affects the shape of the blend shape. Examples of weight for each blend shape for different vowels can be seen in Figure 2. In Figure 2, an example is shown if the audio is using vowels that correspond to the average vowel formant as in Table 1, then the mouth shape will be found in accordance with the combination of three blend shapes that have a certain weight, which affects the mouth shape of the virtual character.

3.2. HMSAM

The HMSAM is a testing model regarding the level of user acceptance of the HMS system. Hedonic motivation systems (HMS) means a more concerned system with the element of pleasure than productivity [6], [7]. Figure 3 shows the HMSAM model [6].

- Perceived ease of use (PEOU) serves to measure the level of ease of use of a system.
- Perceived usefulness (PU) serves to measure the level of performance of the use of a system.
- Curiosity (C) serves to measure the user's curiosity.
- Control (Co) functions to measure the user's perception that he is invited to interact with the system.

- Joy (J) functions to measure the level of pleasure that users get from interactions with the system.
- Behavioral intention to use (BIU) functions to measure the level of a user's desire to use an application.
- Focused immersion (I) functions to measure the user's involvement with the application, which is useful for knowing how deep the user's level of focus is when using the system.

HMSAM was developed to address users' underlying intrinsic motivations in a process oriented context, especially in online gaming, virtual worlds, social network, and gamified learning environments [5], [21]. The HMSAM generally [22], is the combination of Hedonic-motivation system (HMS) [23], perceived ease of use (PEOU), and behavioral intentions to use (BIU), where HMS serves as the critical mediator for the technology acceptance model (TAM) with a concurrent study on user motivation [24].

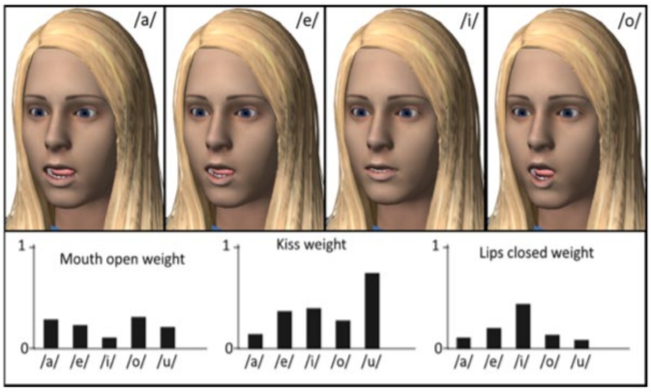


Figure 2. Example of weight of every blend shape for different vowels [1]

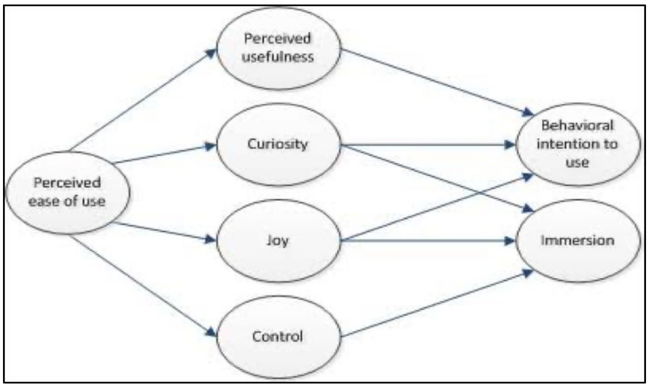


Figure 3. HMSAM model [5]

3.3. Lemeshow formula

The Lemeshow formula is useful to determine the sample size of a population when the exact population size is unknown. The sample size calculation described by the Lemeshow formula is shown in (10). Variable *n* is the sample size, *z* is the confidence score, P is the maximum estimate, and *d* is the precision [25].

$$n = \frac{z_{1-\alpha/2}^{2} P(1-P)}{d^{2}} \quad (10)$$

4. DESIGN AND IMPLEMENTATION

4.1. Design

Jacob's virtual character's design is considered based on Jacob chatbot's specification that has been built before. Jacob uses the English language for input and output. Jacob's virtual character workflow starts from Jacob, where users can ask Jacob's chatbot via the web-based Jacob application. Then, the answer is given through Jacob's desktop-based virtual character. The workflow of Jacob's virtual character is displayed in Figure 4.

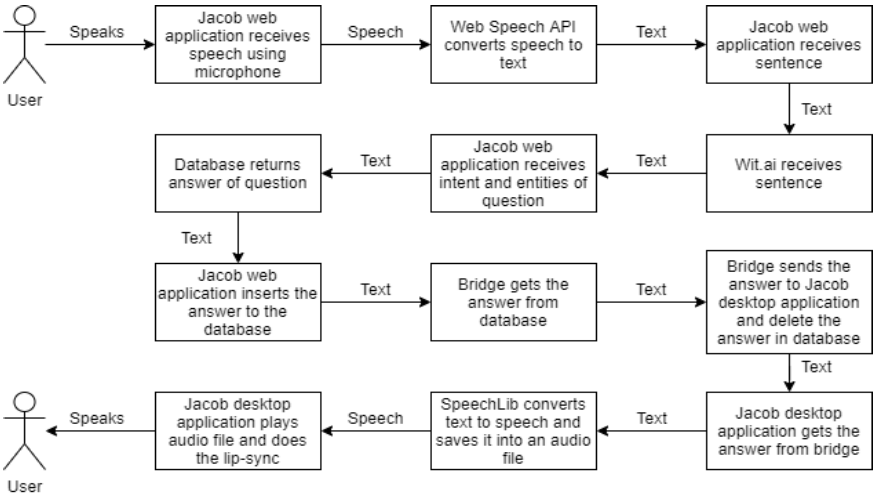

User
Speaks
Jacob web application receives speech using microphone
Speech
Web Speech API converts speech to text
Text
Jacob web application receives sentence
Text
Wit.ai receives sentence
Text
Jacob web application receives intent and entities of question
Text
Database returns answer of question
Text
Jacob web application inserts the answer to the database
Text
Bridge gets the answer from database
Text
Bridge sends the answer to Jacob desktop application and delete the answer in database
Text
Jacob desktop application gets the answer from bridge
Text
SpeechLib converts text to speech and saves it into an audio file
Speech
Jacob desktop application plays audio file and does the lip-sync
Speaks
User

Figure 4. The workflow of Jacob's virtual character

The process starts with user input in the form of a question to Jacob by speech. The question is processed, and the answer to the question is retrieved from the database. This answer is delivered back to the user by speech. The answer is also sent to the virtual character by using a module called Bridge. This Bridge functions as a connection between the Jacob web-based application and the virtual character desktop-based application. It ensures to send the answer to the desktop-based application using the JavaScript object notation (JSON) encode method.

On the Bridge, messages data are also deleted in the database to maintain the storage. Jacob's virtual character receives the message via UnityWebRequest and JSONUtility. After receiving the message in the form of text, the SpeechLib library converts the text message into speech. The speech is saved into an audio file, and the virtual character application plays the audio file and synchronizes the lip movements with the speech. The architectural diagram of Jacob's virtual character is displayed in Figure 5. There are three main components of Jacob's virtual character: the web-based application, the Bridge, and the desktop application.

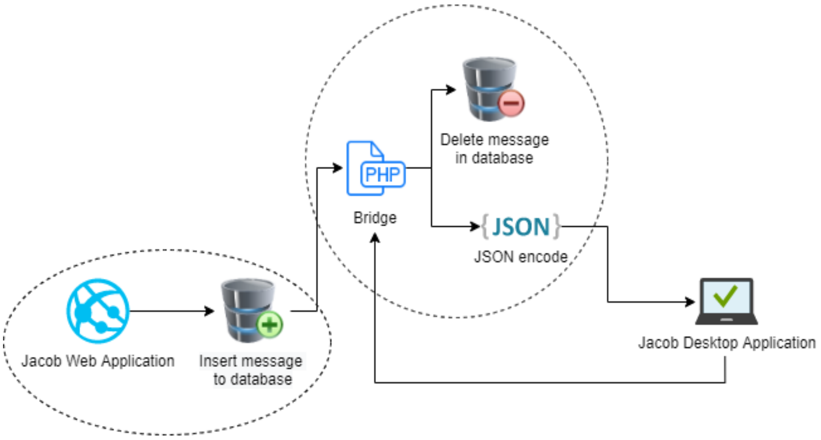

Delete message in database
PHP
Bridge
{JSON}
JSON encode
Jacob Desktop Application
Jacob Web Application
Insert message to database

Figure 5. Jacob's virtual character application's architecture

The desktop application is developed using Unity and C#. The 3D dimensional character has vertices that can be adjusted to the bones of the characters, and blend shapes that can be adjusted for animated movements. Blend shapes can have values within a specific range, and values from blend shapes can be adjusted or encoded. In this research, blended shapes are needed to animation the character's mouth movements when lip-syncing. SpeechLib is used to do speech synthesis for the messages sent by Jacob chatbot web-based applications to be able to output audio or speech on the desktop application. We present the flowchart and pseudocode of the algorithm in Figure 6 and Figure 7, respectively.

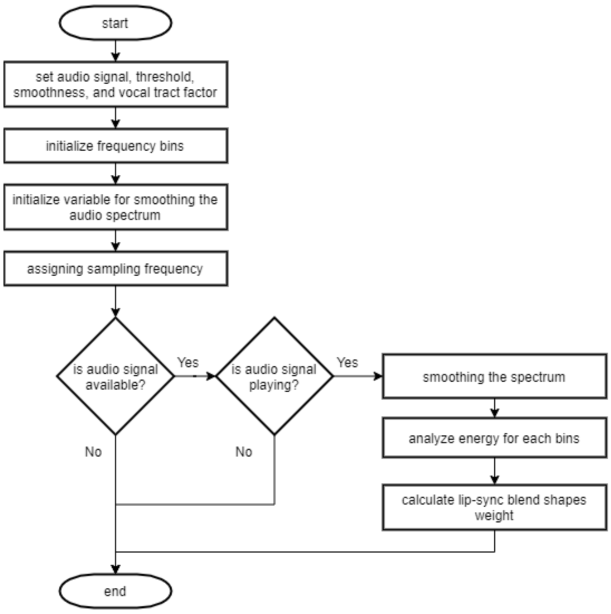

start
set audio signal, threshold, smoothness, and vocal tract factor
initialize frequency bins
initialize variable for smoothing the audio spectrum
assigning sampling frequency
is audio signal available?
Yes
is audio signal playing?
Yes
smoothing the spectrum
analyze energy for each bins
calculate lip-sync blend shapes weight
No
No
end

Figure 6. Rule-based lip-syncing flowchart

```
Initialize audioSignal as audio signal,   as threshold, as vocal tract factor;
Initialize frequency bins:
Fbins[0] ← 0;
Fbins[1] ← 500γ;
Fbins[2] ← 700γ;
Fbins[3] ← 3000γ;
Fbins[4] ← 6000γ;
Initialize as a smoothing variable;
Let Xk be a domain complex frequency;
Let fs as sampling frequency;
Assign the value of fs;
If audioSignal is available:
        If audioSignal is playing:
                Let k be an index of the number of samples per audio block;
                While the number of samples per audio block is not satisfied do
                        Initialize the smoothed spectrum solution:
                        Xk ← τ X-1k+1-τXk;
                        Initialize the smoothed spectrum in dB:
                        Yk ← 20 Xk ;
                EndWhile.
                Let m be an index of the number of frequency bins;
                While the number of frequency bins is not satisfied do
                        Let frequency data indices solution:
                        Findm  ← 2NfsFbinsm;
                        While the spectrum length is not satisfied do
                                Initialize smoothed spectrum in dB that has mapped into dynamic range in an
                                interval [-0.5,0.5]:
                                Yk ← δ+(Y[k]+20)/140;
                                Let sum of each smoothed spectrum that has mapped into dynamic range:
                                Em  ← Em+Yk
                        EndWhile.
                        Update value for energy of each bin solution:
                        Em  ← Em Findm+1-Find[m] ;
                EndWhile.
                Let BSkiss be a kiss blend shape solution:
                If E1<0.2:
                        BSkiss ← 1-2E2 . 5E1;
                Else:
                        BSkiss ← 1-2E2;
                EndIf.
                Let BSlips be lips closed blend shape solution:
                BSlips   ← 3E[3];
                Let BSmouth be mouth open blend shape solution:
                BSmouth ← 0.8(E1-E3);
        EndIf.
EndIf.
```

Figure 7. Rule-based lip-syncing pseudocode

4.2. Implementation

In the rule-based lip-syncing algorithm, the audio signal is processed first so that lip-sync can be performed. Audio signal processing is done by windowing the audio signal using the Blackman window. Then fast fourier transform (FFT) is calculated. This process is done by using the scripting application programming interface (API) from Unity. The three blend shapes that are implemented to the virtual character and user interface of this application are displayed in Figure 8 and Figure 9. The interface, while Jacob is speaking, can be seen in Figure 10.

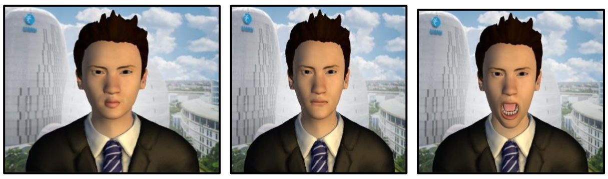
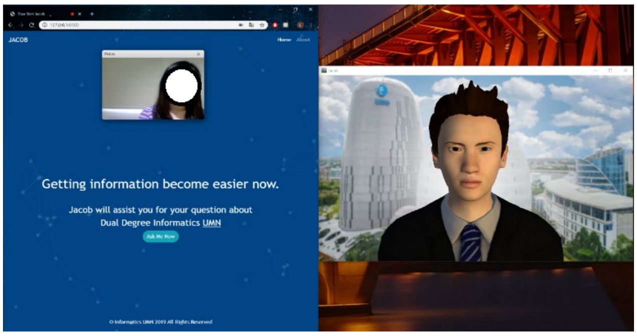


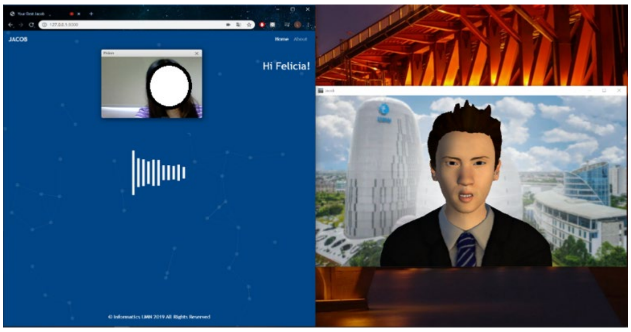
Figure 8. Jacob’s virtual character lips kiss, closed lips, and open mouth



Getting information become easier now.

Jacob will assist you for your question about Dual Degree Informatics UMN

Figure 9. Jacob’s user interface with the virtual character



Hi Felicia!

Figure 10. Jacob’s user interface while speaking

5. RESULTS AND ANALYSIS

Jacob's virtual character is tested with direct interaction between the tester and Jacob. We test the system with live input and different speakers and with several speech audio files. We use a black-box approach to test all functionalities of the system. Evaluation is carried out by gathering user feedback from the questionnaire. The sampling technique used is purposive sampling, in which we aim the participants to be people who are interested in voice chatbots. The population's demography is between 20 and 40 years old, ranging from student, marketing, human resource, software developer, software quality control, and business analyst. Since the population size is unknown, the sample size is determined using the Lemeshow formula, with a z score of 90 percent, a maximum estimate (P) of 0.5, and an error rate of 5 percent. It is obtained from the formula that the minimum sample size required for this study is 67.65.

In this study, seventy people participate in the evaluation process of the application. We use a 6-point likert scale to force choice and yield better results. If a neutral is desired at any point, the “slightly agree” and “slightly disagree” could be averaged together. The 6-point likert scale offers options for completely agree, mostly agree, slightly agree, slightly disagree, mostly disagree, and completely disagree. Table 2 shows the questionnaire questions.

The overall result of the questionnaire can be seen in the chart presented in Figure 11. Figure 12 displays the detailed results of behavioral intention to use. The detailed results of immersion are shown in Figure 13. The application is perceived as easy to use by 89.43%. We found that the control (Co) aspect is the lowest of the four factors, with 76.11%. This is due to the limited control over the virtual character module when talking to the Jacob voice chatbot. The virtual character developed in this study solely perform lip-syncing for the Jacob voice chatbot. There are no other interactions added to the module. However, the application has provided an adequate level of control for the users to decide what to see, what to do, and interactions.

The behavioral intention to use (BIU) is influenced by three factors: perceived usefulness (PU), curiosity (C), and joy (J). We obtained 91.74% for the BIU of the application. However, the focused immersion (I) of the application scores only 72.95%. It is due to the control aspect that influences the focused immersion. Figure 13 shows that most users are absorbed in talking to Jacob with the additional virtual character module. It is also found that the application requires further enhancement to reduce the level of distraction of focus. We found no issues with the usability factor of this work.

Table 2. Questionnaire questions

Questions	
Perceived ease of use (PEOU)	1. Is my interaction with the virtual character Jacob app clear and easy to understand?
	2. Does my interaction with the virtual character Jacob app not require much mental effort?
	3. Do I feel the Jacob's virtual character is free of problems?
	4. Do I find it easy to do what I want to do using the Jacob's virtual character?
	5. Is it easy for me to learn how to operate the Jacob's virtual character?
	6. Is it simple for me to do what I want with the Jacob's virtual character?
	7. Is it easy for me to become skilled at operating the Jacob's virtual character?
	8. Do I find the Jacob virtual character application easy to use?
Perceived usefulness (PU)	1. Does the virtual character take the burden off my mind?
	2. Does the virtual character help me pass the time better?
	3. Does the virtual character provide me with a useful solution?
	4. Does the virtual character help me think more clearly?
	5. Does the virtual character help me feel young again?
Curiosity (C)	1. Does the virtual character delight my curiosity?
	2. Does the virtual character make me curious?
	3. Does the virtual character awaken my imagination?
Control (Co)	1. Do I have much control over the virtual character?
	2. Can I freely choose what I want to see or do?
	3. Do I have little control over what I want to do?
	4. Am I in control?
	5. Do I have no control over my interactions?
	6. Am I allowed to control my interactions?
Joy (J)	1. Do I enjoy the application with the virtual character?
	2. Do I have fun using the Jacob's virtual character?
	3. Is Jacob's virtual character tedious?
	4. Do I find the virtual character annoying?
	5. Do I have great experience of using Jacob with the virtual character?
	6. Does Jacob's virtual character make me dissatisfied?
Behavioral intention to use. (BIU)	1. Do I plan to use the virtual character again in the future?
	2. Will I continue to use the virtual character in the future?
	3. Do I hope to continue using the virtual character in the future?
Focused immersion (I)	1. Can I block other distractions while using the virtual character?
	2. Can I perceive what I am doing?
	3. Am I immersed in the virtual character?
	4. Am I easily distracted by outside interference?
	5. Could My attention not be easily diverted?

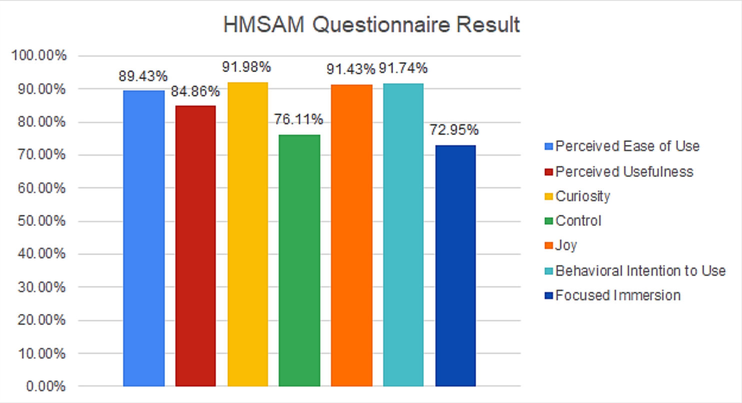


Figure 11. HMSAM questionnaire result

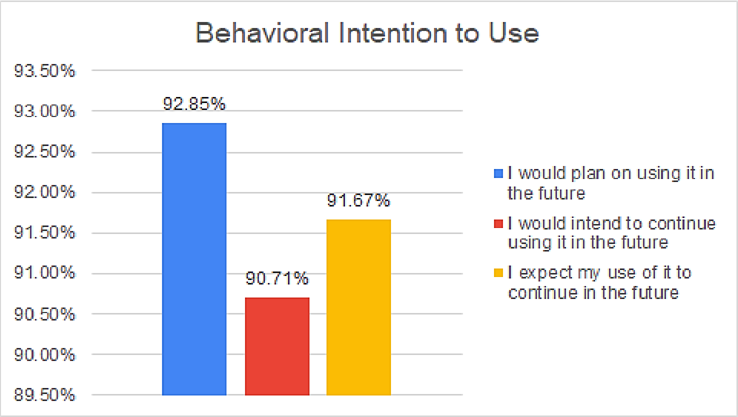


Figure 12. Behavioral intention to use (BIU)

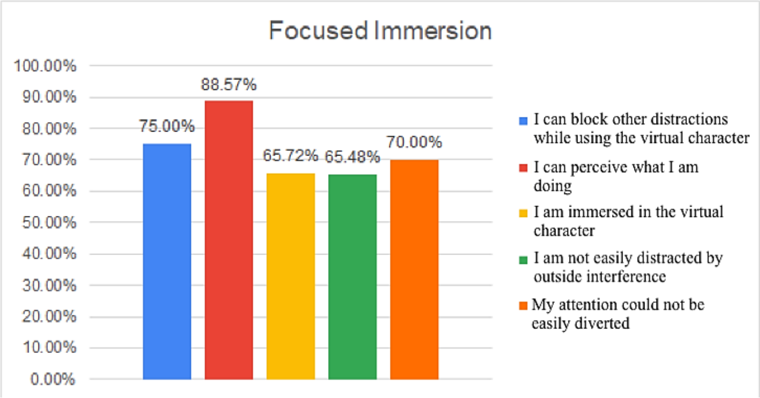


Figure 13. Focused immersion (I)

The overall evaluation score for the application using HMSAM gives a better understanding of the users' perspective on the virtual character's impact on a voice chatbot. This finding supports our research goal to provide a believable virtual character for a voice chatbot application in real-time. However, this approach's limitation is the lack of detailed accuracy on the lip movement and shape of the virtual character during lip-syncing. Other approaches offer higher accuracy but with the expense of extensive training data, effort, and time.

We further analyzed the blend shapes' formulation based on the heuristic approach to enhance lip-syncing accuracy. It is also vital to differentiate blend shapes between male and female virtual characters. We found the most accurate blend shapes formulation for the current virtual character model developed in this study as the following;

$$BS_{kiss} = \begin{cases} (0.34 - E[2]) * 3.6, for\ E[1] \geq 0.2 \\ ((0.5 - E[2]) * 2) * (E[1] * 5),\ for\ E[1] < 0.2 \end{cases} \tag{11}$$

$$BS_{lips} = E[3] * 3 \tag{12}$$

$$BS_{mouth} = 1.15 * (E[1] - E[3]) \tag{13}$$

6. CONCLUSION

This study concludes that the rule-based lip-syncing algorithm with audio-driven articulation and manual mapping is suitable and adequately accurate for synchronizing lip movements with real-time speech. The manual mapping method does not require corpus recording data, making it highly suitable for applications that do not have pre-recorded data.

The approach used in this study also eliminates the necessity of training processes compared to other approaches, such as the neural network. However, the lip-syncing accuracy is lower compared to the data-driven method that used corpus recording data. The Jacob voice chatbot application requires fast lip-syncing in the virtual model's real-time to provide a seamless and believable experience. We measured this by using the HMSAM, and the behavioral intention to use the score of the application is 91.74%, and the focused immersion level is 72.95%. The average user satisfaction score for all aspects (perceived ease of use, perceived usefulness, curiosity, control, joy, behavioral intention to use, and focused immersion) is 85.50%.

Based on this study, the rule-based lip-syncing algorithm provides fast and straightforward design and implementation, including the integration part with the existing Jacob voice chatbot application. We learned that the audio-driven articulation and manual mapping techniques are suitable for real-time applications such as voice chatbot. Although the accuracy level is the trade-off, it is still improvable by using more specific blend shapes.

Future works are to study the tongue's function and importance for a believable virtual character and measure the virtual character in terms of its fluidity and vividness. It is also essential to find the blend shapes' formulation using a more scientific approach instead of a heuristic approach. Including the study of different blend shapes formulation between male and female virtual characters. Comparison with other works is essential to benchmark the approach and implementation chosen in this study.

ACKNOWLEDGEMENTS

The authors would like to thank Universitas Multimedia Nusantara for the support of this study.

REFERENCES

[1] G. Llorach, A. Evans, J. Blat, G. Grimm, and V. Hohmann, “Web-based live speech-driven lip-sync,” *2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, 2016, doi: 10.1109/VS-GAMES.2016.7590381.

[2] S. Wijaya and A. Wicaksana, “JACOB Voice Chatbot Application using Wit.ai for Providing Information in UMN,” *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6S3, pp. 105-109, 2019, doi: 10.35940/ijeat.F1017.0986S319.

[3] A. Archilles and A. Wicaksana, “Vision: A web service for face recognition using convolutional network,” *TELKOMNIKA Telecommunication, Computing, Electronics and Control,* vol. 18, no. 3, pp. 1389-1396, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14790.

[4] Octavany and A. Wicaksana, “Cleveree: An artificially intelligent web service for Jacob voice chatbot,” *TELKOMNIKA Telecommunication, Computing, Electronics and Control,* vol. 18, no. 3, pp. 1422-1432, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14791.

[5] D. Oluwajana, A. Idowu, M. Nat, V. Vanduhe, and S. Fadiya, “The adoption of students’ hedonic motivation system model to gamified learning environment,” *J. Theor. Appl. Electron. Commer. Res.*, vol. 14, no. 3, pp. 156-167, 2019, doi: 10.4067/S0718-18762019000300109.

[6] P. B. Lowry, J. E. Gaskin, N. W. Twyman, B. Hammer, and T. L. Roberts, “Taking ‘fun and games’ seriously: Proposing the hedonic-motivation system adoption model (HMSAM),” *J. Assoc. Inf. Syst.*, vol. 14, no. 11, pp. 617-671, 2013, doi: 10.17705/1jais.00347.

[7] M. J. Kim and C. M. Hall, “A hedonic motivation model in virtual reality tourism: Comparing visitors and non-visitors,” *Int. J. Inf. Manage.*, vol. 46, pp. 236-249, 2019, doi: 10.1016/j.ijinfomgt.2018.11.016.

[8] J. Cassell *et al.,* “Animated conversation: Rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents,” *Proc. 21st Annu. Conf. Comput. Graph. Interact. Tech. SIGGRAPH 1994*, no. May 2014, 1994, pp. 413-420, doi: 10.1145/192161.192272.

[9] I. Poggi, C. Pelachaud, F. de Rosis, V. Carofiglio, and B. De Carolis, “Greta. A Believable Embodied Conversational Agent,” *Multimodal Intelligent Information Presentation*, 2005.

[10] G. Zoric and I. S. Pandzic, “A real-time lip sync system using a genetic algorithm for automatic neural network configuration,” *2005 IEEE International Conference on Multimedia and Expo*, 2005, doi: 10.1109/ICME.2005.1521684.

[11] G. Llorach and J. Blat, “Say hi to eliza: An embodied conversational agent on the web,” *Lecture Notes in Computer Science*, 2017, doi: 10.1007/978-3-319-67401-8_34.

[12] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher, “Synthesizing obama: Learning lip sync from audio,” *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1-13, 2017, doi: 10.1145/3072959.3073640.

[13] S. L. Taylor, M. Mahler, B. J. Theobald, and I. Matthews, “Dynamic units of visual speech,” 2012.

[14] J. Wolfe, M. Garnier, and J. Smith, “Vocal tract resonances in speech, singing, and playing musical instruments,” *HFSP J.*, vol. 3, no. 9, pp. 6-23, 2009, doi: 10.2976/1.2998482.

[15] P. Podder, T. Zaman Khan, M. Haque Khan, and M. Muktadir Rahman, “Comparative Performance Analysis of Hamming, Hanning and Blackman Window,” *Int. J. Comput. Appl.*, vol. 96, no. 18, pp. 16891-6927, 2014, doi: 10.5120/16891-6927.

[16] S. Scherer, G. M. Lucas, J. Gratch, A. Rizzo, and L. P. Morency, “Self-Reported Symptoms of Depression and PTSD Are Associated with Reduced Vowel Space in Screening Interviews,” *IEEE Trans. Affect. Comput.*, vol. 7, no. 1, pp. 59-73, 2016, doi: 10.1109/TAFFC.2015.2440264.

[17] J. Liu, M. You, C. Chen, and M. Song, “Real-time speech-driven animation of expressive talking faces,” *Int. J. Gen. Syst.*, vol. 40, no. 4, pp. 439-455, 2011, doi: 10.1080/03081079.2010.544896.

[18] P. Jourlin, J. Luettin, D. Genoud, and H. Wassner, “Acoustic-labial speaker verification,” *International Conference on Audio- and Video-Based Biometric Person Authentication*, 1997, pp. 319-326, doi: 10.1007/bfb0016011.

[19] Y. Xu, A. W. Feng, S. Marsella, and A. Shapiro, “A practical and configurable lip sync method for games,” *MIG '13: Proceedings of Motion on Games,* 2013, pp. 131-140, doi: 10.1145/2522628.2522904.

[20] A. Wang, M. Emmi, and P. Faloutsos, “Assembling an expressive facial animation system,” *Sandbox '07: Proceedings of the 2007 ACM SIGGRAPH symposium on Video games*, 2007, pp. 21-26, doi: 10.1145/1274940.1274947.

[21] D. Lie, J. C. Young, S. Hansun, “Dietary Application For Diabetic Patients Using Gamification Method,” *International Journal of Scientific & Technology Research.*, vol. 9, no. 4, 2020.

[22] A. A. Alalwan, Y. K. Dwivedi, N. P. Rana, B. Lal, and M. D. Williams, “Consumer adoption of Internet banking in Jordan: Examining the role of hedonic motivation, habit, self-efficacy and trust,” *Journal of Financial Services Marketing*, vol. 20, no. 2, 2015, doi: 10.1057/fsm.2015.5.

[23] J. Martí-Parreño, E. Méndez-Ibáñez, and A. Alonso-Arroyo, “The use of gamification in education: a bibliometric and text mining analysis,” *J. Comput. Assist. Learn.*, vol. 32, no. 6, pp. 663-676, 2016, doi: 10.1111/jcal.12161.

[24] H. Van Der Heijden, “User acceptance of hedonic information systems,” *MIS Q. Manag. Inf. Syst.*, vol. 28, no. 4, pp. 695-704, 2004, doi: 10.2307/25148660.

[25] S. Lemeshow, D. W. Hosmer Jr, J. Klar, and S. K. Lwanga, “Part 1: Statistical Methods for Sample Size Determination,” *Adequacy Sample Size Heal. Stud.*, 1990, doi: 10.1186/1472-6963-14-335.

BIOGRAPHIES OF AUTHORS

Felicia Priscilla Lovely is a Software Developer in Nexsoft Indonesia who started her career as Full Stack Developer, and currently working as Front End Developer. She earned Bachelor Degree of Computer Science from Universitas Multimedia Nusantara on 2020.

Arya Wicaksana is a lecturer at the Department of Informatics at UMN. He received Master Degree in VLSI Engineering from Universitas Tunku Abdul Rahman. He successfully demonstrated the UTAR first-time success ASIC design methodology on a multi-processor system-on-chip project using 0.18um processing technology in 2015. His main research interests are quantum computing, hardware/software co-development, and computational intelligence. He recently worked on a human-like voice chatbot system called Jacob and post-quantum cryptography for blockchain applications. He is affiliated with ACM and IEEE as a professional member. He has served as an invited reviewer in IEEE ACCESS, IJNMT, and IFERP and an invited author in IntechOpen and other scientific publications.