Lightweight Multi-objective Voice Adaptation for Real-time Speech Interaction Applied in Games

1st Mads Midtlyng
Department of Computer Science
Hosei University
Tokyo, Japan
midtlyng.madsalexander.9c@stu.hosei.ac.jp

2nd Yuji Sato
Department of Computer Science
Hosei University
Tokyo, Japan
yuji@k.hosei.ac.jp

***Abstract*— This paper proposes a novel voice adaptation method that we applied to interactive activities such as games where source and target data are unaligned. Conventional methods have seen the use of probabilistic models or more recently, Deep Neural Networks. Common for most methods is that they require multiple subjects to train in conjunction, thus voice adaptation is not practical to be used in commercial applications. We propose a method which convert audible frequencies to light spectrum simple RGB color format, and not comparing sound signal similarities, but rather likeness in color. The comparison is done using multi-objective optimization which considers raw and normalized frame colors as two separate objectives to be evaluated, respectively audible and spectral structure. The distance for the objectives is used to select an ideal output frame. Finally, prosodic information such as speech intensity is translated from measured input values onto the designated output frame. The method is evaluated using MOS, ABX, performance benchmark and lastly implemented into the Unity3D game engine as a proof of concept. Results show good sound quality and high performance with little output fragmentation.**

Keywords—voice adaptation, speech processing, real-time, multi-objective optimization problems, video games

I. Introduction

Voice adaptation (VA), sometimes referred to as voice conversion or voice transformation [1-5], is the act of translating a message spoken in a source voice into a target voice, retaining the speaking pattern represented in prosodic information. This effectively allows the user to speak with another subject's voice through a microphone and processing component. VA can be categorized into two types: Parallel and nonparallel. The former trains two or more subjects to map speech characteristics and sometimes physical traits. This process can be long and complicated, making it difficult for anyone to use such a system without lengthy preparations. However, parallel VA is the most used approach. The latter is when the source and target speakers are inheritable unrelated, allowing the training to be done by one subject, and the actual VA by another. This can be achieved by either constructing a pseudo dataset for each source and target speaker pairs, transformation by using existing parallel datasets with separate utterances paired by estimation, or lastly by estimating corresponding phonemic content per speaker.

This paper introduces a novel nonparallel method with features to enhance adaptation quality, ease of use and hope to see real world use. Our method is designed to work with very limited amount of training data, opening the possibility that anyone can train and use such a system. We have learned a great deal from previous [6-7] research and applying various methods in order to reduce frame dependency between speakers. This functions by normalizing both voices to a common format, based on unaligned data where frames can be looked up for correspondence during the adaptation process. This paper proposes many improved measures to solve the primary challenges with VA, as such our method consists of two steps; pre-processing in order to create a target voice profile from phonemic content, and real-time matching which compares frames from the source and target. The inner working of these steps is detailed in section 3.

In traditional online games, a player may choose an alias as their name and an avatar as their visual representation, but their voice is always transmitted as their own. We think VA is a fitting element to introduce into online games as a way to test the technology, and so that players may not only feel more anonymous behind another voice, but to increase game-world immersion by having the players speak with the voice of their game character. Currently, voice-interactable software do not feature VA components. The state of VA technology has been immature for commercial use.

In order to create a reliable and performant VA system, we can presume two primary requirements: One; it must be able to match frames from two speakers with high accuracy, and two; it must be able to relay the prosodic information from the source to the target frame. from other methods is that we shift the focus from viewing VA as traditional speech processing problem that handles the human physique, to a search problem that views a subject's voice as a collection of the sounds they can produce. Therefore, our goal with pre-processing is to obtain the possible sounds the target can produce and store this information, then in real-time puzzle together the desired uttering from comparing frames using multi-objective optimization. Additionally, instead of comparing acoustic information, we translate frames to RGB color values, simplifying the format that the multi-objective optimization function needs to work with. It is much easier to compare two colors using a distance function than raw or even processed acoustic information for every frame, thus improving the real-time performance of the overall method.

II. Related Work

A. Traditional Voice Adaptation

Most traditional research that were considered state-of-the-art are based on probabilistic models and rule-based methods. Generally, spectral conversion techniques have been used for over a decade, and they can usually perform more accurate spectral mapping using Gaussian Mixture Modelling (GMM) compared to the rule-based methods which can be unstable in their results. Probabilistic models have been somewhat of a go-to approach [3] [8-9] for voice

adaptation. Many concurrent systems focus on the spectral conversion of the adaptation, and often apply basic adjustments such as pitch shifting to simulate prosody [8] [10].

B. *Later Voice Adaptation*

Not only statistical methods have seen the light such as Wu et al. [11] and Takashima et al. [12] where they employ exemplar-based voice conversion. They saw an improved conversion quality comparable to Maximum Likelihood GMM (ML, GMM). Additionally, Chen et al. [13] uses a layer-based generative training Deep Neural Network (DNN) to perform voice conversion. They saw experimental improvements over classical methods. However, the training is complicated and requires multiple source speakers. Other research [14] employs a training step using interactive evolution which considers the parameters of pitch, power and length. These are then subsequently applied to real-time adaptation in order to perform prosody. Results show that evolutionary computation could get closer to a target compared to a human with trial-and-error. [15]'s approach considers mapping of the voice spectrum which is stored in a "codebook", and then codebooks between speakers are compared. In its learning step, they employ two speakers and dynamic time warping in order to produce vectors that could be corresponding between them. In [16], they employ VOCALOID's database which is created in an associated way as our voice profiles, except here, too, the physical aspects of the voice were taken into consideration. In even more recent times, more research has attempt nonparallel rather than parallel VA despite its difficulties using various methods such as *CycleGAN* [17] and spectral conversion [18]. There have been international contests [19] to judge several VA systems, but none have seen commercial use in interactive software.

Many recent works use variational auto-encoders (VAE) which is a deep learning technique. VAEB (Bayes) is one of the more popular versions of this technique and is used to learn a model p using an encoder q. p is parametrized as (1) where $\vec{\mu}(z), \vec{\sigma}(z)$ are further parametrized by a neural network. Finally, the model q is similarly parametrized.

$$p(x|z) = \mathcal{N}\left(x; \vec{\mu}(z), diag\left(\vec{\sigma}(z)\right)^2\right)$$
$$p(z) = \mathcal{N}(z; 0, I), \qquad (1)$$
$$q(z|x) = \mathcal{N}\left(z; \vec{\mu}(x), diag\left(\vec{\sigma}(x)\right)^2\right)$$

III. Design and Supporting Methods

Design-wise our method is very different from previous work and related work. The reason we chose to implement multi-objective optimization problems (MOP) (see section 4) is due to the fact that non-parallel VA, in various approaches has difficulties finding the ideal unaligned target data. MOP can help us decide which target frame are ideal based on several factors that does not need to be pre-trained. Our method takes a signal processing problem and turns it into a more traditional search problem; the search is assisted by multi-objective evolutionary algorithms which is a powerhouse of decision-making when the overall problem is large, but can be divided into many smaller instances. There are two steps to conducting VA; pre-processing and frame adaptation. Before any VA can be conducted, a voice profile must be generated in the pre-processing step as seen in Fig. 1.

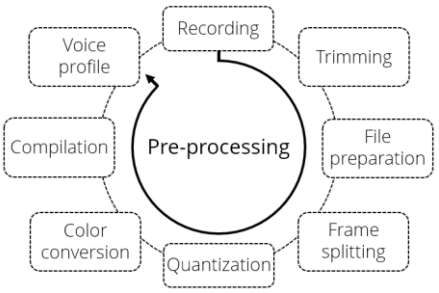

Fig. 1. Pre-processing steps to produce a voice profile.

This is done by recording a voice subject who is guided by a user-friendly graphical user interface in order to complete the recording within desired parameters. The subject is presented with a manuscript generated by the system for a given language, which is a text composed of supporting words, phonemes and various combined utterings. Grammar and semantics are irrelevant, we are only trying to produce unique sounds for the recording. The system makes sure the subject speaks within "default" intensities, e.g. the intensity for normal, relaxed speech. Failure to utter within the desired parameters prompt a re-uttering until the recording is completed. At the end, the voice recording is trimmed for empty noise before compiled into an optimized file which is used for later output sampling. Next, the acoustic data is split into many 6 milliseconds long frames with a splitting increment of 3 milliseconds, generating large volumes of data. From here on, any processing takes place on a frame-to-frame basis.

A. *Stylized Quantization*

In order to perform reliable comparisons between various frames, they must all be normalized to a base format. This reduces frame dependency between speakers, allowing Speaker A's components for 'hello' to be recognized in Speaker B's 'hello' by eliminating miniscule acoustic indifferences which otherwise would have an impact. It should be noted that the output frames are most likely collected from various uttering's fragments, rather than the direct correlating word, due to the fact we record all the phonemes used in various utterings. Quantization is often used in traditional signal processing, here we apply a stylized version which uses pitch as the primary parameter and constraints are set for min and max fundamental frequencies in which the frame is bound to.

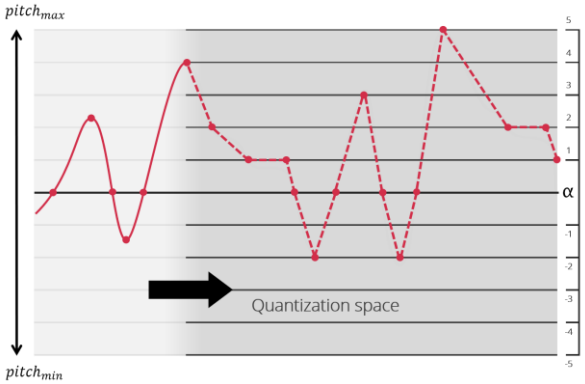

Fig. 2. Acoustic data is quantized according to a resolution alpha.

$$Q(x) = \Delta \cdot \left(\frac{x}{\Delta} + \frac{1}{2}\right) = \Delta \cdot floor\left(\frac{x}{\Delta} + \frac{1}{2}\right) \qquad (2)$$

The frame's polarizing amplitude points are mapped to resolution steps, but points with very small distance between each other are discarded. The frame is forced into the quantization space regardless of its specifications. The resolution is called alpha and dictates the complexity of the quantized signal. A low value yields an abstracted signal, while a high value is truer to the initial acoustic representation. The quantizer is non-uniform and uses rounding (2) to the nearest alpha step. The step size is denoted as Δ, and the notation floor() depicts a floor function. Sampling rate is dictated by adverse values in the quantization space (Fig. 2) that are considered paramount features of the frequency domain, mapped at runtime for each frame. There is no reconstruction stage and the results are used for the final frame generation.

B. Frame Generation

Sound and color can both be defined by a frequency, thus it is possible to convert (3) between them for exchanging representations of data. The frequency is just one aspect of their value, the other is represented by smaller components. In sound these components can be constrained to pressure and time, while in color they are red, green and blue values. Having to account for the relationship in sound pressure in the temporal domain for the length of the frame is inconvenient as complexity and voice profile volume increases. With color we only need to worry about 3 static values for any frame in any circumstance. When a frame is translated to color, we extract the average frequency for the entirety of the frame based on an *n* number of points, for two different states of the frame. The first state is before quantization, where the points are any polarizing amplitudes and crossing the baseline. The second is the same for the quantized frame. These make up the objectives in our multi-objective function which checks the distance (4) between two frame's colors, so that we can find the best match between the actual acoustic representation and abstracted representation. The RGB values are clamped from the output frequency converted in (3) to the frequencies associated with the visible light spectrum, from violet (790 THz), blue, cyan, green, yellow, orange to red (405 THz). The min and max values for sound are human hearing range, which varies from as low as 12Hz to 28,000Hz, but is mostly described as 20Hz to 20,000Hz, also used in this paper.

$$Lv_{out} = \left(\frac{Sv_{in} - Sv_{min}}{Sv_{max} - Sv_{min}}\right) \times (Lv_{max} - Lv_{min}) + Lv_{min} \quad (3)$$

$$dist = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2} \quad (4)$$

IV. Multi-objective Voice Adaptation

Multi-objective optimization problems considers optimization problems consisting of more than one objective function that are to be optimized simultaneously. This is a tool often used in fields such as engineering and economics in order to find an optimal solution for conflicting objectives. The output of the optimization is not a single perfect solution, but many distributed over a Pareto front (Fig. 3). An ideal solution can be selected from the set depending on which function holds more weight.

Previously we attempted to find the ideal output frame based on a single objective, and while it in many cases does find an acceptable solution, it could not do so for all. The human voice is a difficult medium because it is composed of many fine traits varying from each person. To improve the matchmaking of source-to-target frames, we evaluate multiple objectives for a single frame to find the target which most accurately represents the source sound. This is a novel use of MOP as it has not been used for VA previously. The strength of MOP is that it is more likely reach ideal results the longer it runs. However, with real-time output there is a serious time-constraint, as such there are strict parameters for how our implementation can run, as seen in Table 1. Since we are using a standard function such as DTLZ2, we must convert our objective values to its parameter space using (3), but with DTLZ2's known min and max objective values from a reference run.

A. Multi-objective Evaluation using MOEA/D

In MOEA/D [20], multi-objective problems are handled as a set of single-objective problems, each defined by a scalar function utilizing a set of weight vectors (5). The normalization of the objective space for m-objectives can be defined as (6) subject to $x \in X$, where $f_i(x)$ is the *i*-th objective to be minimized $(i = 1, 2, \dots, m)$, x is a decision vector and X is a feasible region of x in the decision space. A set of uniformly distributed weight vectors are generated according to (7), and the amount of weight vectors *H* are the same as the population size. Each single-objective problem with its own weight vector has a single solution, and the goal is to search for the best solution along each weight vector.

$$w = (w_1, w_2, \dots, w_m) \quad (5)$$

$$Minimize\ f(x) = \big(f_1(x), f_2(x), \dots, f_m(x)\big) \quad (6)$$

$$\sum_{i=1}^{m} w_i = 1 \ and\ 0 \le w_i \le 1\ for\ i = 1, 2, \dots, m$$
$$w_i \in \left\{0, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H}{H}\right\}\ for\ i = 1, 2, \dots, m \quad (7)$$

We run two instances of MOEA/D for two frames in a queue. This effectively creates a minimum output lag of 12 milliseconds plus-minus the time it takes to initialize MOEA/D for each instance's new frame, however the result outweighs this delay as we can very accurately find ideal target frames. In Fig. 3 the two objectives are represented as f_1 and f_2, f_1 being normalized frame color and f_2 raw frame color with the distribution of 15 voice profile frames as solutions. The initial selection of which frames from the profile will be used in the optimization is decided by their primary normalized color value. The 15 frames with the closest normalized color value to the current input frame are inserted into the MOEA/D instance. The reason we limit this to 15 is because population size has a large effect on the time MOEA/D spends on a single generation. Depending on the system's hardware, this can be modified to accommodate for performance. In Fig. 4 we have a resulting Pareto Front from frame colors. The objective $distance_{normalized}$ calculates the frame color distance converted to objective spaced based on the normalized frame's features, while $distance_{raw}$ the same for the raw frame. An input frame's raw frame color and normalized color are expected to vary slightly, as the normalization simplifies the signal to get rid of unwanted miniscule features. By using these are our objectives, we are able to evaluate between actual auditory features and simplified features (necessary for comparing very diverse vocal samples) using weighted color distance. If we wish to

consider both evaluations equally important, we can select a solution closest to the middle of the objective min- and max values. Often the case of short runtimes based on few generations or short amounts of time, MOEA/D struggles to find optimal solutions for the whole population. However what we are left with are much better options than assuming ideal frames on our own and the configuration of MOEA/D can be changed at any time depending on the user's hardware, done in an instant rather than hours or days recalibrating with large datasets.

TABLE I. MOEA/D CONFIGURATION

Parameter	Value
Problem type	DTLZ2
Max generations	10
Test function	TCHn1
(or max time alottment)	6ms
Population size	15
Neighborhood size	3
Concurrent instances	2 threads

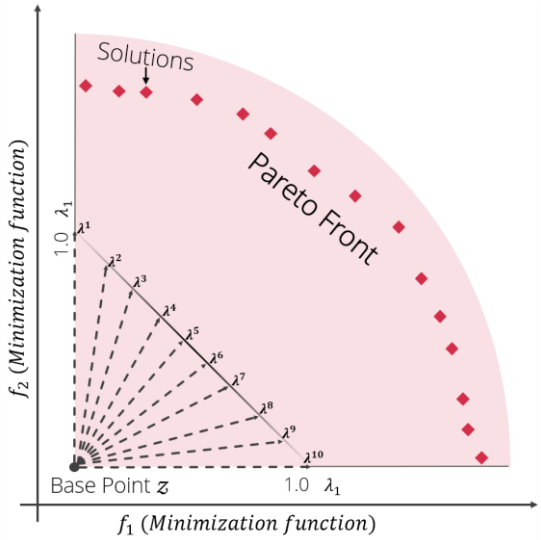

Fig. 3. Reference Pareto Front for 15 solutions over 100 generations.

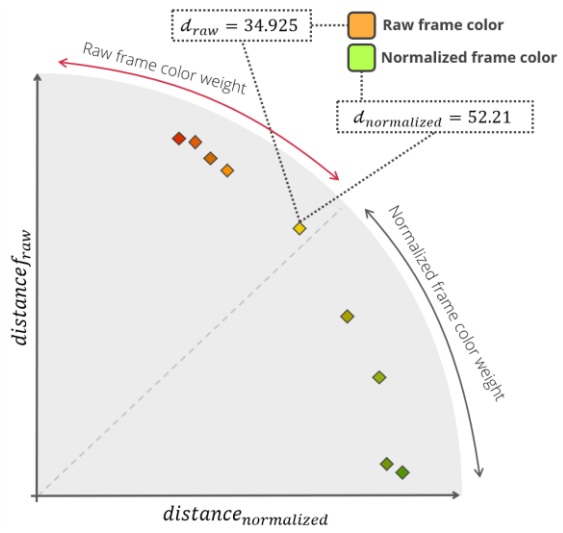

Fig. 4. Using weights we can decide which feature is more important for final selection by distance *d*.

B. *Sound to Color Effectiveness*

By translating a sound to color, we achieve the fact that the format's memory footprint is smaller due to the simplified container structure, allowing for better performance. It's also easier to visualize the process so that we can create better interactions. A color can be normalized to a single integer value, very convenient in MOEA/D which can achieve more desirable results the longer it runs. Thus, the simpler data conversion the better.

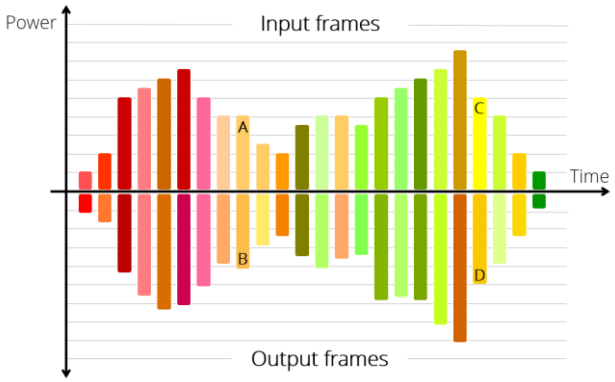

Fig. 5. Frame colors for input to optimimal target frames.

In modern games, especially A.I. components require many CPU resources to perform. If this technology is to be utilized in games, we know it must be very lightweight for developers to consider it. In Fig. 5 we see bars representing the intensity and color signature of input and output frames. For frames A and B, their colors are 94.19% similar, while C and D are 87.27%, which we consider acceptable. If we pre-process even longer, we are more likely to achieve color similarity above 90% for most cases.

C. *Partial Frame Smoothing*

As each output frame is obtained from varying locations in the voice profile, their temporal structure does not seamlessly connect where one frame ends and the next starts. This affects final sound quality and to solve this we propose partial frame smoothing (Fig. 6) using weights for a portion of the frame. It's a simple, but effective solution. By not doing so the stitching of random frames together would sound obvious to the listener as oddly sounding artifacts throughout the output stream. We apply this to the first and last sections of a frame, using the information obtained from the previous frame.

D. *Dynamic Power*

In order to present the output speech with the prosodic depiction of the input speaker, dynamic power is used where per source frame a power level is measured from -1.0 to 1.0 and multiplied to the target frame (Fig. 7) before output. This is possible due to the nature of how we create our voice profile which is based off a "neutral" power level derived from the intensity of normal speech, ranging from 50 to 60 decibels. We can obtain the current power of a frame by analyzing its frequency and the average volume distribution, from where the decibel levels are obtained, which are in turn normalized to fit in the range of -1.0 to 1.0 based on realistic min- and max values for speaking and the range of the input device.

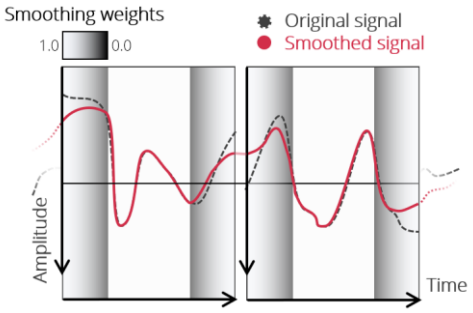

Fig. 6. Unrelated frames are smoothed together for a more natural sound (applied after dynamic power adjustment).

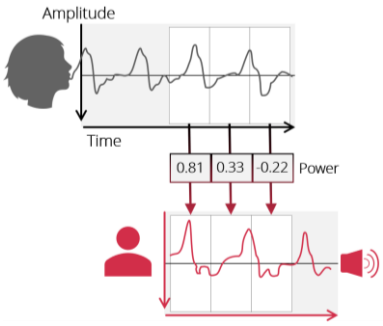

Fig. 7. Source power is translated onto the target frame.

E. Technical Implementation

While this essentially is a problem of sound, visual representation is equally important in order to provide good usability. Using .NET Core 3.1 we have a custom software which is used to define manuscripts, create voice profiles and perform tests.

TABLE II. CONTENTS OF A VOICE PROFILE FRAME

Field	Value
ID	12503
Audio begin	3m24s115ms
Audio end	3m24s121ms
Raw frame color	rgb(23, 55, 210)
Normalized frame color	rgb(19, 68, 197)
Average pitch (normalized)	0.4211
Base power (normalized)	0.325

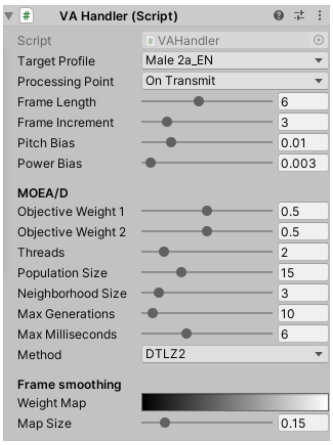

Fig. 8. Unity3D VA-Handler component parameters.

The same components used in this are seamlessly importable to the Unity3D [21] (Fig. 7) game engine. The implementation of MOEA/D is a custom C# version based on the source code used for the research done by [20]. Porting of this framework allows for high effectiveness when testing and using the VA system.

V. EXPERIMENTS AND RESULTS

We wish to evaluate the degree of sound quality, adaptation similarity, ease of training, ease of use and implementation to a game engine. The tests were judged by both native and non-native English speakers using a voice profile based on one English manuscript with a duration of 5 minutes. The speakers were all 10 males (ages 22-35), none with a background in speech processing. The speech material to be adapted was based on different content than the source manuscript. Fragmentation is the case where a frame could not be adapted and is presented in the performance test results. Adaptation quality and similarity is judged in the Mean Opinion Score (MOS) test. A blind test (ABX) was also evaluated. In the game engine application (Table 4), the test manuscript was used to input speech which is adapted through the software, sent over the network and back again before outputted in the target voice profile to work as a proof of concept that VA can be utilized in online game software. The test environment is presented in Table 5.

A. Methods

1) MOS: Measures the quality of the VA output. The score represents the overall stimuli and is expressed as an absolute category rating (ACR) as seen in Table 3. A voice subject different from the voice profile uttered words from a testing manuscript and the output played back to individuals who rated the experience.

2) ABX: A subject is presented with two audio samples, A and B and must be able to differentiate between them. The samples represent the target profile's original audio and source speaker's adapted frames. Finally a Sample X is presented which can be either of the previous, and the order of A and B's playback is random. The subject must be able to identify sample X as the A or B with a high probability. A low probabilty means that the adapted playback is similar to the original audio. Each set of A-B are played back 5 times for 20 different utterings. So that the subject does not to become accustomed to the same sounds thus "blinded", the utterings are also played in a random order.

3) Performance: The software is benchmarked in terms of how quickly a target frame can be acquired and how close the color distances are. Additionally, missing frames are registered as fragmentation. The test was carried out for 20 words of various length uttered 3 times in random order.

TABLE III. ABSOLUTE CATEROGY RATING FOR MOS

Rating	Degree
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

TABLE IV. GAME ENGINE AND PROJECT SETTINGS

Field	Type
Game engine	Unity 3D (v. 2019.1)
Lisence type	Free
Project type	Universal Render Pipeline (3D)
Input system	Unity3D native
Primary components	Input, AudioClip, Network VAHandler[a]

[a] Custom component. Others are native.

TABLE V. TEST ENVIRONMENT

Component	Specification
OS	Windows 10 64-bit edition
CPU	Intel Core i5-4690K 3.5-3.9 GHz
GPU	ASUS STRIX RX480 8GB
RAM	16 GB
Input device	Dynamic microphone at 44.1 kHz/48 kHz sample rate
Software	.NET Core v3.1 application

B. Results

1) MOS: Playback of 30 selected utterings.

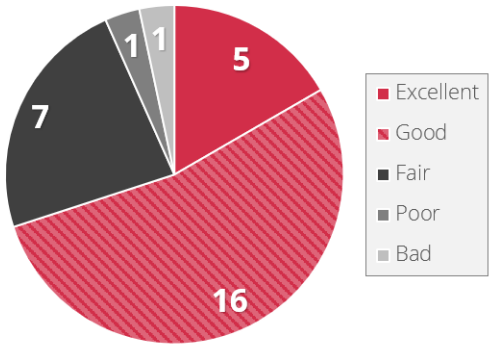

Fig. 9. MOS rating: Majority rated output as good.

Generally, the shorter utterings were rated the best, but not exclusively.

2) ABX: Results for 30 utterings over 5 sessions.

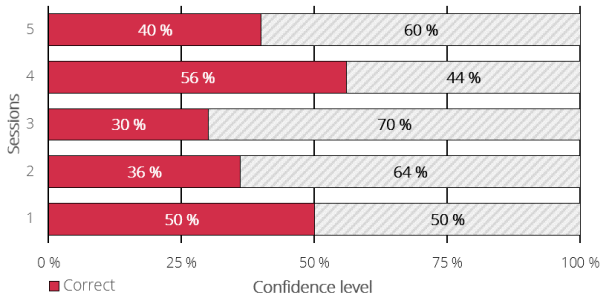

Fig. 10. Blind test: Sound quality is comparable to original audio.

Lower percentages of correct guesses conclude that the subject was unable to tell if the playback was adapted or the original voice.

3) Performance: Speeds and output fragmentation.

TABLE VI. PERFORMANCE, AVERAGE AND WORST

Type	Average	Worst
Input-to-Color	380µs	590µs
MOEA/D Init per frame	940µs	4ms
Input to match	5	11ms
Post-processing (power, smoothing)	2ms	4ms
Timer bias (best guess)	2ms	4ms
Fragmentation per 100 frames	5	11

MOEA/D initialization time varies every time as it uses a random function to evaluate the first generation before beginning its main evaluation loop. In a best-case scenario, the target frame was found within the time it was processing the next frame in the queue, effectively having an output delay of only one frame. Fragmentation could be registered as 0, but we chose not to accept the results from MOEA/D if the color distance was greater than 150 (max distance is 441).

VI. DISCUSSION

A. Comparing Results to Recent Related Work

Comparing to related work in the field, [17] who are using a method known as CycleGAN (Generative Adversarial Networks) used for a non-parallel based off 200 utterances per speaker and unaligned data. CycleGAN is 6-layer forward neural network (NN) which in a MOS test achieved slightly above average results, better than traditional work, and on pair with concurrent research who use NN. Additionally, GANs on a GPU may take hours, and on a common CPU more than a day depending on the dataset and number of layers. However, MOS' scoring offers little insight into nuanced perceptions of the user; thus, a more detailed rating system is ideal, but since MOS is commonly used in the field it is the practical for comparing research. We imagine players or small-time developers would not see this as user-friendly unless the results can be excellent. Another use of neural networks in [22] is based on DNN and while they observe good results in the MOS test, the execution time is not great because it includes time spent extracting features, converting and restoration. One of our goals is to propose a lightweight, speedily set up VA framework for both the creator and user, and we can avoid such calibration steps by utilizing MOP in our non-parallel method. [23] employs more training time and larger data and claim to have a superior approach, but their results suggest typical VA performance near the average mark. If we can propose a faster method that achieve better than average results using a more lightweight training approach, DNN might be more suited for non-parallel VA methods where there are more constraints for data alignment. There is however a trade-off between having a simple to use system and sound quality. We think this can be solved by several simple improvements for our case, such as even shorter frames, introducing relation-conscious population selection for MOEA/D, and improved manuscript reading. One issue is what if our manuscript recording did not contain proper phoneme information. How can we reconstruct the missing frames in our library, could it be done during

runtime? Possibly it could be plausible to stitch together assumed missing frames from even smaller segments of existing frames, based on known information. [24] employs frequency warping using many warping factors but considers going the opposite route of using just a single warping factor. Their results are considered good average performance compared against the state-of-the-art statistical methods at the time. These results show that current approaches all fall towards the average mark, even the more state-of-the-art methods such as DNN. [25] utilizing DNN see mixed results regarding speaker likeness and training volume, as such this is a global challenge for VA. The proposed approach is also very portable and new voices can easily be added. It has few dependencies so that it functions without pre-formatted speech data for training. [26] investigated at the time a new VA method based on frequency warping plus amplitude scaling, which while required parallel data, not much of it. They compared it to state-of-the-art (GMM) methods, while also claiming to be a simple framework. They saw that such a method had very slight improvements, but they made a point that their method is more robust than traditional methods in the case where training data is not in ideal amounts. Their subjective evaluations saw average results.

In the experimental tests, testing software built using [27] was used, while in a more commercial scenario this would have been seamlessly integrated into the game software with very little required configuration from the source speaker, as preparation can be done ahead when creating the target profile. It should ideally be as natural to setup as regular voice chat is today. The user should not be required to learn a complex procedure in order to benefit from this system.

B. Benefit of MOP in VA

As mentioned, the voice is a difficult medium to work with. Despite our ears being able to identify 'hello' as 'hello' for 100 different utterings, its digital signature is different for each one due to fine nuances in the spectral domain. By erasing these with stylized quantization, we're left with many comparable data, but the voice has more than one feature, thus it's difficult to compare all data fairly and decide which attributes are more important. MOP is a powerful tool in exactly this arena and by changing the problem perspective from acoustic to search problem, we are truly benefitting of MOP in this case as we can easily dictate the attributes used for objectives. Compared with other state-of-the-art methods like DNN, a framework like MOEA/D can easily be tweaked to accommodate for new tactics by a few parameters while a trained DNN might have to re-learn with new acoustic material from the start.

By using both a raw and normalized frame, we can perform weighted selections based on which objective should be considered more important. On one side, comparing raw frames singlehandedly is often difficult as perfect matches are not guaranteed, which is why normalized frames are used to hint the objective function in the right direction. However, relying solely on the normalized frames could result in matches that look similar in color, but are slightly off in auditory senses due to using average frame pitch for converting to colors. Using the weighted combination of these gives the most reliable target frame for a given input frame. We realize could be further improved by subdividing a frame and use the average pitch for a given set of sub-frames and is considered for future work.

Compared to our previous work [7] which did not use MOP, but a custom search implementation based on cross-correlation of frame abstractions, we observe two major improvements. First, since we are dealing with a more effective data format, comparisons yield more robust results so that fragmentation technically does not occur. If we wish to accept these results as ideal or not is another factor. Secondly, if we previously failed to find a satisfactory frame, the search would span across a much larger data set and spend a large amount of time (in the real-time sense) before it returned an empty frame. Now, we initialize MOEA/Ds population with 15 of the already closest frames to our target, thus the range of frames in which we search is narrowed down, and it is guaranteed to return a result which immensely speeds up the process. Right now, each frame is handled independently, but we consider that using the previous frame's information for populating the next MOEA/D instance, we can improve the frame selection to even more relevant frames from the voice profile.

C. Role of VA in Future Game Software

We believe a technology like VA can potentially be accommodated with other voice-utilizing software such as Text-to-Speech systems or A.I.-based assistants found in smart devices in the future. There is also a risk that such a technology may be used to impersonate other individuals, however where this technology could be used for its full good potential is in the online gaming world, where players might miss out on experiences because they are too shy to talk with other players, or revealing their voice in general. This can be solved by introducing VA to the game and will not only protect the identity of the player but increase game-world immersion as the output voice may be that of the chosen character or import a custom profile from an online library to customize the game further. Such a feat would complete the anonymity of an online virtual presence. Depending on the game type, various complexity of voice profiles could be implemented. Shorter ones for simple voice commands (e.g. "Supplies, please"), and longer for free speak. Thus, VA could potentially be applied in games ranging from strategy to open-world. For smaller studios, rather than dealing with costly voice-over work, a simple recording from a voice subject could suffice for the entirety of the game's lifespan if done satisfactory, and the library could easily be added to with minimal work.

VII. Conclusion

We proposed a very simple to use lightweight voice adaptation which introduces multi-objective optimization problems in MOEA/D to find an ideal target frame from unaligned data in real-time. Due to this, performance and matching rate is very reliable, and this process could be utilized by both developers and users in order to create voice profiles based on relatively short recordings. The increased performance opens possibilities for other refinements in future work that will focus on improving the current design to accommodate for shorter training times and matching quality.

REFERENCES

[1] Y. Eason, Stylianou, "Voice Transformation: A Survey," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Taipei, pp. 3585-3588, April, 2009.

[2] D. Erro, A. Moreno, "Weighted Frequency Warping for Voice Conversion," 8th Annual Conference of the International Speech Communication Association INTERSPEECH, Antwerp, pp. 1965-1968, August, 2007.

[3] Y. Stylianou, O. Cappé, E. Moulines, "Continuous probabilistic transform for voice conversion," IEEE Transaction on Speech and Audio Processing, vol. 1, Seattle, pp. 285-288, May 1998.

[4] T. Toda, H. Saruwatari, K. Shikano. "Voice Conversion Algorithm based on Gaussian Mixture Model with Dynamic Frequency warping of STRAIGHT spectrum," Proc. ICASSP, pp. 841-844, 2001.

[5] E. Moulines and Y. Sagisaka, "Voice conversion: State of the art and perspectives," Speech Communication. Special Issue, vol. 16, no. 2, February 1995.

[6] M. Midtlyng, and Y. Sato, "Real-time Voice Adaptation with Abstract Normalization and Sound-indexed Based Search," IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, pp. 60-65, October 2016.

[7] M. Midtlyng, and Y. Sato, "Voice Adaptation from Mean Dataset Voice Profile with Dynamic Power," IEEE International Conference on Systems, Man, and Cybernetics (SMC), Shizuoka, pp. 2037-2042, October 2018.

[8] A. Kain and M.W. Macon, "Spectral Voice Conversion for Text-To-Speech Synthesis," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seattle, pp. 285-299, May 1998.

[9] H. Ye and S. Young, "Quality-enhanced Voice Morphing using Maximum Likelihood Transformations," IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 4, pp. 1301-1312, July 2006.

[10] Y. Chen, M. Chu, E. Chang, J. Liu and R. Liu, "Voice Conversion with Smoothed GMM and Map Adaptation," 8th European Conference on Speech Communication and Technology (Eurospeech 2003 – Interspeech 2003), Geneva, pp. 2413-2416, September 2003.

[11] Z. Wu, T. Virtanen, E. S. Chng and H. Li, "Exemplar-based sparse Representation with Residual Compensation for Voice Conversion," IEEE Transaction on Audio, Speech, and Language Processing, vol. 22, no. 10, pp. 1506-1521, October 2014.

[12] R. Takashima, T. Takiguchi and Y. Ariki, "Exemplar-based Voice conversion in noisy environment," IEEE Spoken Language Technology Workshop (SLT), Miami, pp. 313-317, December 2012.

[13] F. Villavicencio and J. Bonada, "Voice Conversion using Deep Neural Networks with Layer-wise Generative Training," IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP), Journal, pp. 1859-1872, December 2014.

[14] Y. Sato, "Voice Quality Conversion using Interactive Evolution of Prosodic Control," Applied Soft Computing Journal, Elsevier, pp. 181-192, June 2004.

[15] M. Abe, S. Nakamura, K. Shikano and H. Kuwabara, "Voice Conversion Through Vector Quantization," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 1, New York, pp. 655-658, April 1988.

[16] F. Villavicencio and J. Bonada, "Applying Voice Conversion to Concatenative Singing-Voice Synthesis," 11th Annual Conference of the International Speech Communication Association (INTERSPEECH), Chiba, pp. 2162-2165, September 2010.

[17] F. Fang, J. Yamagishi, I. Echizen, and J. Lorenzo-Trueba, "High-Quality Nonparallel Voice Conversion Based on Cycle-Consistent Adversarial Network," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, pp. 5279–5283, September 2018.

[18] C-C. Hsu, H-T. Hwang, Y-C. Wu, Y. Tsao and H-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Jeju, pp. 1–6, December 2016.

[19] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen et al., "The Voice Conversion Challenge 2018: Promoting Development of Parallel and Nonparallel Methods," Odyssy 2018, April 2018.

[20] Q. Zhang, H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," IEEE Transactions on Evolutionary Computation, vol. 11, issue 6, pp. 712–731, December 2007.

[21] Unity3D, Unity Technologies. Accessed on: January 1. 2019 [Online]. Available: https://unity.com/.

[22] Y. Sekii, R. Orihara, K. Kojima, Y. Sei, Y. Tahara and A. Ohsuga, "Fast Many-To-One Voice Conversion using Autoencoders," International Conference on Agents and Artificial Intelligence (ICAART), Porto, pp. 164-174, Feburary 2017.

[23] G. Kotani, D. Saito and N. Minematsu, "Voice Conversion Based on Deep Neural Networks for Time-variant Linear Transformations," Asia-Paficic Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, pp. 1259-1262, December 2017.

[24] M. Tamura, M Morita, T. Kagoshima and M. Akamine, "One Sentence Voice Adaptation using GMM-based Frequency Warping and Shift With a Sub-band Basis Spectrum Model," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, pp. 5124-5127, May 2011.

[25] Y. Li, K.A. Lee, Y. Yuan, H. Li and Z. Yang, "Many-to-many Voice Conversion Based on Bottleneck Features with Variational Autoencoder for Non-parallel Training Data," Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Hawaii, pp. 829-833, August 2018.

[26] D. Erro, E. Navas and I. Hernáez, "Parametric Voice Conversion Based on Bilinear Frequency Warping Plus Amplitude Scaling," IEEE Transactions on Audio, Speech, and Language Processing, vol 21, No. 3, pp. 556-566, March 2013.

[27] Microsoft .NET Core 3.x SDK (2019) [Online]. Available: https://dotnet.microsoft.com/download/dotnet/current