USER-GUIDED VARIABLE-RATE TIME-STRETCHING VIA STIFFNESS CONTROL

Nicholas J. Bryan, Jorge Herrera, and Ge Wang

Center for Computer Research in Music and Acoustics
Department of Music, Stanford University
California, USA
{njb, jorgeh, ge}@ccrma.stanford.edu

ABSTRACT

User control over variable-rate time-stretching typically requires direct, manual adjustment of the time-dependent stretch rate. For time-stretching with transient preservation, rhythmic warping, rhythmic emphasis modification, or other effects that require additional timing constraints, however, direct manipulation is difficult. For a more user-friendly approach, we present work that allows a user to specify a time-dependent stiffness curve to warp the time axis of a recording, while maintaining other timing constraints, such as a desired overall recording length or musical rhythm quantization (e.g. straight-to-swing), providing a notion of stretchability to sound. To do so, the user-guided stiffness curve and timing constraints are translated into the desired time-dependent stretch rate via a constrained optimization program motivated by a physical spring system. Once the time-dependent stretch rate is computed, appropriately modified variable-rate time-stretch processors are used to process the sound. Initial results are demonstrated using both a phase-vocoder and pitch-synchronous overlap-add processor.

1. INTRODUCTION

Most straightforward audio time-stretch processors operate using a single constant overall stretch rate or stretch factor. The stretch factor Γ corresponds to the output-to-input length of a time-stretch process and the stretch rate A corresponds to the inverse of the stretch factor. When a higher degree of musical control is needed, variable-rate methods that allow warping of time axis are used, as in the case of time modification with transient preservation [1], rhythmic warping (e.g. straight-to-swing [2]), rhythmic emphasis modification, automatic dialogue replacement, and other creative time modification effects. Several high-quality processing techniques can be used for this purpose and include phase-vocoders, (pitch) synchronous overlap-add processors, or sinusoidal modeling with resampled envelopes. In most cases, these algorithms are controlled automatically with little-to-no user input or completely manually by directly manipulating the time-dependent stretch rate $\alpha(t)$ or stretch factor $\gamma(t)$. For situations that require fine-tuning or other user input, however, both fully automatic and completely manual control can be very difficult or impossible to maneuver.

By design, fully automatic methods have no mechanism for user input and are of limited utility to scenarios that require additional control. In contrast, direct manipulation of the stretch rate offers complete control, but is usually very arduous to maneuver. This is particularly evident when performing constrained time modifications, which require variable-rate stretching in conjunction with other timing constraints, such as a global overall stretch rate, rhythmic quantization, or similar. In this case, the time-varying stretch rate (or factor) is coupled with the overall length or similar constraints, making direct manipulation of the stretch rate difficult and thus requiring significant effort to achieve a desired result.

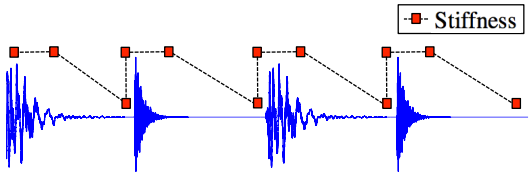

Figure 1a: *Variable-rate time-stretching via user-guided stiffness control. The proposed method can stretch an input sound to a fixed length, while locally warping the time axis to preserve onsets with minimal user input.*

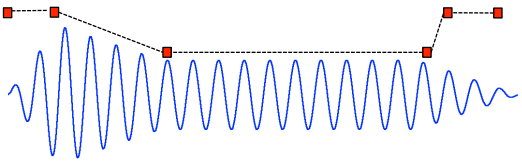

Figure 1b: *User-guided stiffness can allow an attack, decay, sustain, and release of a note stretch differently from one another, given an overall desired length.*

To overcome this issue, we propose a method that allows a user to independently specify a time-dependent stiffness curve, or oppositely stretchability curve, along with timing constraints. The user input is then automatically translated into a time-dependent stretch rate using a constrained optimization program. The output of the process yields an optimal time-dependent stretch rate which is then used to generate the desired variable-rate effect using any suitably modified pre-existing time modification algorithm. The metaphor of stiffness is directly inspired by a physical spring chain system and allows a user to specify how each region of sound is stretched relative to one another. This idea is first illustrated in Fig. 1a, where four eighth-notes are annotated with a stiffness curve in an effort to preserve the note attack. When stretched to a desired length or stretch factor, low-valued stiffness regions are stretched more than high-valued regions in a smooth way, while the rhythm is maintained. Alternatively, Fig. 1b shows an additional application where the attack, decay, sustain, and release of a note are annotated with varying stiffness values, allowing each region to stretch differently given a desired overall length (attack

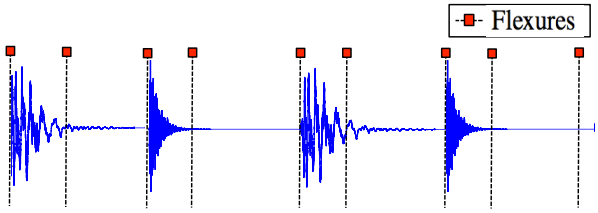

Figure 2: *One-dimensional flexure interface for variable-rate time-stretching control. Moving a marker modifies the stretch rate of the two neighboring regions.*

and release are stretched less than decay and sustain).

A full description of the proposed method including initial formulation and extensions is discussed in §3, preceded by a brief outline of past work in §2. Then, a short discussion regarding the phase-vocoder and pitch-synchronous overlap-add processor implementations used are discussed in §4, followed by results and conclusions in §5 and §6 respectively.

2. PAST WORK

Past work involving user-guided variable-rate time-stretching control is found in many commercial audio editing software systems such as ProTools, Logic Pro, FL Studio, and others. In a number of cases, fully automatic algorithms are parameterized with several options for the user to guide time-stretch algorithms. User guidance can be as simple as denoting whether the content to be processed is rhythmic, monophonic, or wide-band audio, or somewhat more involved such as allowing users to edit note onset markers used for transient preservation and rhythmic quantization. For situations that require time-warping effects, however, further control is needed.

In these cases, recent interfaces, such as found in the work of Nielson and Brandorff [3], LogicPro, or ProTools improve upon direct manipulation of the stretch rate and allow users to input pivot or flexure points to easily manipulate a time-dependent stretch rate as shown in Fig. 2 and Fig. 3 (the former is a one-dimensional version of the latter). To correct a mistaken note onset, for example, a user can place two flexure points around a given note to isolate a region and then move a third point in the middle to push and pull the audio content on either side. The corresponding stretch rates on either side are then computed as step-wise constant values satisfying the timing constraints defined by the outer flexures.

While pivot or flexure point control provides a significant improvement over direct manipulation of the stretch rate, certain aspects leave room for improvement. Firstly, when attempting to perform non-trivial stretch modifications over time, such as linearly slowing down a region of a recording while maintaining an overall desired overall length, the step-wise constant nature of the flexure manipulation can require a large number of flexure points and significant user effort. Secondly, it is unclear how each flexure region should behave when a time-warped recording is re-stretched to an updated length. The most straightforward way would be to equally stretch all regions proportional to their respective region length given the overall desired length, but when a user wishes to preserve transients or similar, they will potentially have to completely re-annotate a large number of flexure points. Because of this, it is desirable to supplement the capabilities of such interfaces

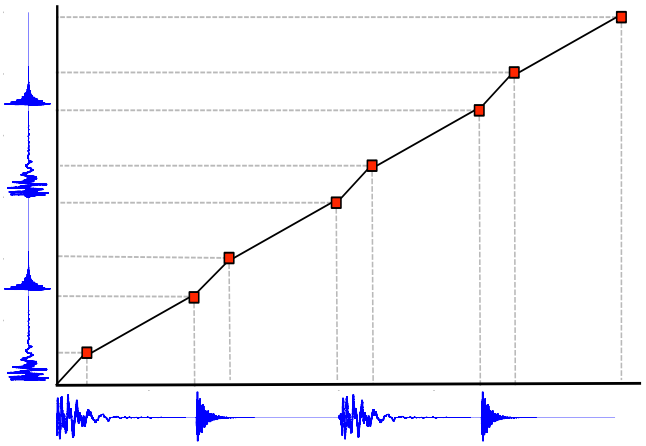

Figure 3: *Two-dimensional flexure interface. Similar to Fig. 2, moving a marker modifies the stretch rate of the two neighboring regions.*

with an ability to specify how each region of sound is stretched proportional to one another within a single framework as proposed in this work.

3. PROPOSED METHOD

To accommodate adequate control over variable-rate time-stretch algorithms as discussed above, we require the ability to allow a user to specify how to stretch every portion of a recording relative to each other, while being able to specify an overall stretch length or satisfy similar constraints, such as flexure points. To do so, we first employ the metaphor of a physical spring system with an initial formulation discussed in §3.1. The initial formulation is then subsequently modified to accommodate more intuitive control over the resulting time-dependent stretch rate, requiring the initial formulation to be posed as a cost function to be minimized as discussed in §3.2. After the minimization formulation, extensions that allow rhythmic warping and smoothing of user input are discussed in §3.3. For consistent notation, all formulations are written as convex optimization or feasibility problems and solved using `CVX`, a package for specifying and solving convex programs [4, 5]. For specific details of numerical optimization techniques used, the interested reader may refer to [6].

3.1. Initial Formulation

As mentioned, our initial formulation is motivated by a physical chain of ideal springs. The idea is that a given audio recording can be divided up into N regions or blocks. Each block is then associated with an ideal spring and corresponding stiffness coefficient within a chain of springs as shown in Fig. 4a. A user can then control the non-negative spring stiffness coefficients k_i which in turn modifies the displacement x_i of each spring from its respective equilibrium position. Fig. 4b shows two springs with equal stiffness coefficients, while Fig. 4c shows the same springs with different stiffness coefficients. Lower stiffness results in a greater ability to stretch or compress, while a higher stiffness results in a lesser ability to stretch or compress and allows a user to specify

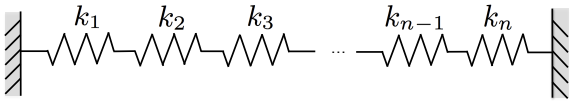

Figure 4a: *Chain of ideal springs with varying stiffness.*

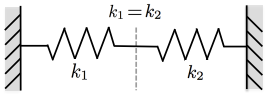

Figure 4b: *A two-spring system with $k_1 = k_2$, $x_1 = x_2$.*

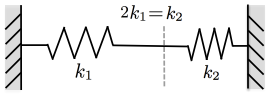

Figure 4c: *A two-spring system with $2k_1 = k_2$, $x_1 = 2x_2$.*

how to proportionally stretch each region of sound, relative to one another.

The modified lengths are computed by solving a system of linear equations as a result of Hooke's law $F_i = -k_i x_i$ invoked by each spring, Newton's third law, and a specified overall length. We can formally write this in closed form, or for consistency reasons, as a linearly constrained feasibility problem

$$\begin{aligned} \text{find} \quad & \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{f} = \mathbf{0} \\ & \boldsymbol{x}^{\mathrm{T}}\mathbf{1} = L \end{aligned} \tag{1}$$

where $L = \Gamma \cdot L_0$ is the desired output length in seconds, L_0 is the original length of the given recording, $\boldsymbol{x} \in \mathrm{R}^N$ is the vector of spring length displacements from equilibrium in seconds, $\boldsymbol{f} \in \mathrm{R}^{N-1}$ is a vector of force equalities with element $f_i = k_{i+1}x_{i+1} - k_i x_i$ for $i = 1, 2, .., N-1$, and $\mathbf{0}$ is a vector of zeros of length $N-1$. Intuitively, this says find a vector of N spring length displacements that satisfy the required force equations and sum to the desired output length. Once the spring lengths are calculated, the time-dependent stretch factor is computed element-wise as the ratio of the resulting spring length displacements over reference displacement lengths for the same system with uniform stiffness values and $L = L_0$.

In certain cases, however, unintuitive results can occur in this formulation because each spring has a natural length of zero. More specifically, this formulation does not allow an unprocessed recording to be considered to have zero force on the springs, implying the recording is always stretched from zero length. To allow a non-zero natural length and provide more intuitive control, we must modify Eq. (1) to accommodate a natural length $\boldsymbol{x}_0$ for each spring and add additional inequality constraints to enforce non-negativity on the individual spring lengths via

$$\begin{aligned} \text{find} \quad & \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{f} = \mathbf{0} \\ & (\boldsymbol{x} + \boldsymbol{x}_0)^{\mathrm{T}}\mathbf{1} = L \\ & \boldsymbol{x} + \boldsymbol{x}_0 \geq \mathbf{0} \end{aligned} \tag{2}$$

This says find a vector of N spring displacement values that are greater than $-\boldsymbol{x}_0$, satisfy the required force equalities, and sum to the desired output length. In most cases, $\boldsymbol{x}_0 = L_0/N \cdot \mathbf{1}$ with $\mathbf{1}$ equal to a column of ones of length N. Consequently, when L is greater or less than L_0, the recording is stretched or compressed respectively. When $L_0 = L$, all forces are equal to zero and no processing is done as desired by intuition. In the case where $L_0 = L$ and warping based on the annotated stiffness coefficients is still desired, the initial length $\boldsymbol{x}_0$ can be manipulated to stretch and/or compress the sound while maintaining a constant length.

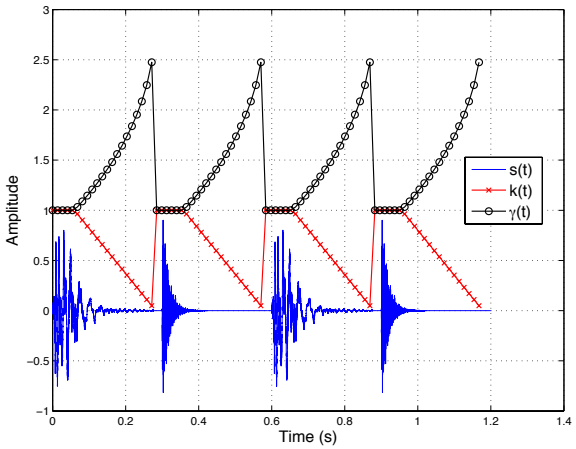

Figure 5: *Conversion of user-guided stiffness to variable-rate stretch factor constrained by a global stretch factor of $\Gamma = 1.5$ and weighting $\mu = .01$ using Eq. (3).*

3.2. Minimization Formulation

Because of the non-negativity constraint $\boldsymbol{x} + \boldsymbol{x}_0 \geq \mathbf{0}$, Eq. (2) can potentially become infeasible. To overcome this issue, we can reformulate the system as an optimization problem that attempts to minimize a cost function, rather than satisfy hard equality constraints from the spring forces. To do so, a linearly constrained least-squares problem is used to penalize the norm of the force constraint vector and spring displacement lengths, resulting in

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & ||\boldsymbol{f}||_2 + \mu||\boldsymbol{x}||_2 \\ \text{subject to} \quad & (\boldsymbol{x} + \boldsymbol{x_0})^{\mathrm{T}}\mathbf{1} = L \\ & \boldsymbol{x} + \boldsymbol{x_0} \geq \mathbf{0} \end{aligned} \tag{3}$$

where μ represents a weighting factor. This formulation attempts to find a vector of N spring length that sum to the desired length L, are non-negative, and minimize the norm of the approximate derivative (first-order difference) of the individual forces of the spring chain. Intuitively, this is equivalent to finding spring lengths that disturb the force equalities the least, with an additional penalty on the spring displacement lengths to regularize the solution. The resulting spring lengths are then converted into the time-dependent stretch factor as the ratio of the resulting spring lengths over the natural length. An example of processing the user-guided input of Fig. 1a into a time-dependent stretch rate given a global stretch factor and a fixed value of μ is shown in Fig. 5 (for further results and interpretation, see §5).

If desired, alternative cost functions can be used in replace of $||\boldsymbol{f}||_2$ which result in similar, but distinct outcomes. Two such alternatives include the norm of the individual forces of the spring chain (i.e. $||\mathbf{k} \circ \boldsymbol{x}||_2$, where $\circ$ is the Hadamard product) and the norm of the approximate derivative of $\boldsymbol{f}$. The former more directly approaches the problem of constrained variable-rate control and does not necessarily require the metaphor of spring stiffness,

while the latter corresponds to penalizing the curvature of individual forces of the spring chain. For simplicity, however, we limit further extensions and results to the main minimization formulation of Eq. (3).

3.3. Extensions

Additional extensions to the minimization formulation can allow for improved results and further applications. Such extensions include time-dependent upper bounds on the stretch factor, smoothing of coarse user input, and rhythmic warping effects, such as straight-to-swing manipulation. These effects are accomplished by carefully adding cost terms and/or additional constraints to Eq. (3). For a time-dependent upper bound on stretch factor, an additional set of inequalities can be added analogous to the non-negativity constraint of the spring lengths (e.g. $\boldsymbol{x} + \boldsymbol{x}_0 \leq 2\boldsymbol{x}_0$ for a maximum spring length of twice the natural length). For smoothing of coarse user input, additional regularization terms can be added, such as a penalty on the second derivative or curvature of the spring lengths similar to cubic smoothing splines. For rhythmic warping effects, additional equality constraints can be used to enforce a desired rhythm.

For a specific example of straight-to-swing warping of a recording of four straight quarter notes (i.e. Fig. 1a), we can add three additional sets of equality constraints to Eq. (3), resulting in

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & ||\boldsymbol{f}||_2 + \mu||\boldsymbol{x}||_2 \\ \text{subject to} \quad & (\boldsymbol{x} + \boldsymbol{x_0})^{\mathrm{T}}\mathbf{1} = L \\ & \boldsymbol{x} + \boldsymbol{x_0} \geq \mathbf{0} \\ & (\boldsymbol{x^1} + \boldsymbol{x_0^1})^{\mathrm{T}}\mathbf{1} = \tfrac{2}{3}L/2 \\ & (\boldsymbol{x^2} + \boldsymbol{x_0^2})^{\mathrm{T}}\mathbf{1} = \tfrac{1}{3}L/2 \\ & (\boldsymbol{x^3} + \boldsymbol{x_0^3})^{\mathrm{T}}\mathbf{1} = \tfrac{2}{3}L/2 \end{aligned} \tag{4}$$

where the $\mathbf{1}$ is a vector of ones of the appropriate length and the superscripts $1, 2, 3$ denote the spring displacement lengths that correspond to the first, second, and third quarter of the recording (additional constraints for the fourth region are not needed). The additional constraints force the first and third eighth notes to occupy two-thirds time of a quarter note and the second and fourth eighth notes to occupy one-thirds time of a quarter note. The stiffness control then dictates how the sound is stretched internal to each eighth note region. Overall, the optimization formulation allows for a number of extensions and is suitable for many interesting musically motivated modifications. Future extensions could even go about learning the stiffness values from data and suggest how a user should stretch or compress a sound for a given recording.

4. TIME-STRETCH IMPLEMENTATIONS

Once the user-guided stiffness input is converted into a time-varying stretch factor for a given recording, a time-stretch processor must be used to actually process the sound according to the variable-rate control. To show that the proposed stiffness control easily integrates into existing time-stretch processors, both phase vocoder (PV) and pitch synchronous overlap add (PSOLA) processing methods are used. As the method of control is the focus of this work and not the time-stretch processors themselves, however, only basic implementations are used.

For the variable-rate phase-vocoder implementation, two open source implementations were tested and include [7] and [3]. The former operates by computing a constant hop size Short-Time Fourier Transform (STFT) of a signal, constructing a modified STFT by linearly sampling the magnitude and updating phases of the input STFT for a given time path, then inverting the modified STFT signal for the final result. The latter implementation operates by computing the STFT with a non-uniform hop size, constructing a modified STFT with updated phase values, and then inverting the result with a constant hop size STFT. In both cases, the time-varying stretch factor given by the optimization stage is then used as an input control signal on a block-by-block basis to the phase vocoder. If the discretization of the optimization stage does not correspond to the hop size of the STFT required for the phase vocoder (typical), linear interpolation of the control signal can be used to align the signals appropriately.

For the PSOLA algorithm, an implementation as described in [8] was developed. The pitch detection is performed using the YIN algorithm [9] and the pitch marking is computed via a slightly modified version of the algorithm described in [8]. The time-varying stretch factor $\gamma(t)$ output by the optimization stage is then sampled at the non-uniformly spaced pitch mark location dependent on the input signal, which inform the algorithm of the correct local stretch factor. Because of the non-uniformity, linear interpolation of $\gamma(t)$ is again required similar to the PV processor.

5. RESULTS

To illustrate how the proposed stiffness control method performs on several example situations, we first apply Eq. (3) with various parameterizations to the recording and user input of Fig. 1a. Fig. 5 shows a baseline result with a global stretch factor of 1.5 times the original length, a fixed value of $\mu = .01$ (as seen before), and N chosen to adequately capture the detail of the user annotation. As illustrated, a few simple user input automation points result in a stretch factor signal $\gamma(t)$ that preserves regions of the audio signal $s(t)$ with high stiffness $k(t)$, stretches regions with low stiffness, and maintains the required constraint of Γ, all in a smooth way. Moreover, once the stiffness points are input to the system, the user can freely update the overall length without having to reset the stiffness values.

For further intuition on how the various parameters effect the resulting output, §5.1 illustrates the effect of varying the overall stretch length, §5.2 shows the effect of the regularization parameter μ, and §5.3 displays an example of straight-to-swing modification with stiffness control. After that, a final example of modifying a speech recording for rhythmic emphasis modification is given to show potential usefulness for automatic dialogue replacement, where an actor's voice is manipulated to emphasize certain syllables for dramatic effect, while maintaining a fixed overall length. Overall, the results show that the proposed method is able to achieve intuitive control over variable-rate time-stretching. To listen to sound examples, please see `http://ccrma.stanford.edu/~njb/research/stretch/`

5.1. Varying Stretch Length

To illustrate how Fig. 5 is effected by changing the overall recording length, Fig. 6 shows varying overall stretch factors Γ with a fixed $\mu = .01$. When Γ increases, the stretch factor signal $\gamma(t)$ increases in slope, but maintains the same relative shape. When Γ decreases, the overall slope of $\gamma(t)$ again follows suit, but only

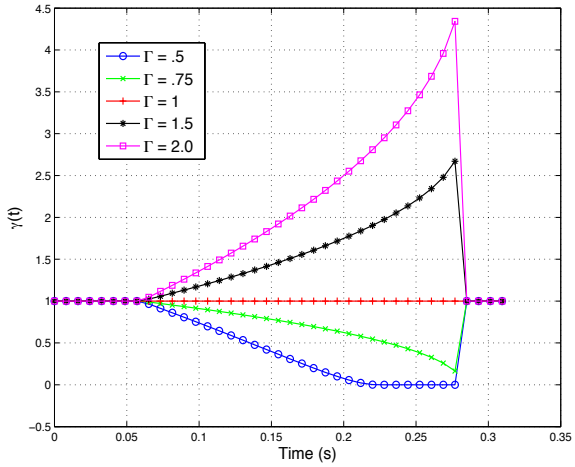

Figure 6: *Results of varying the overall global stretch factor* Γ. *Only one-fourth of the audio signal displayed in Fig. 5 is shown.*

up to a saturation point when the region cannot compress any further. In this case, the optimization program simply tries its best to uphold users' expectations from the given stiffness input, but still satisfies the overall length constraint as required. Without the minimization formulation, the saturation would cause infeasible solutions. Also note the shape of the stretch factor is dependent on the regularization weight parameter μ.

5.2. Effect of Regularization

To show the effect of varying the regularization weight parameter, Fig. 7 displays the results of varying values of μ for the first one-fourth of Fig. 5. We can notice that when μ is very small or zero, the optimal solution stretches only a small number of the least costly segments a great deal. Such a result causes an almost glitch-like effect and is not musically pleasing in most situations, justifying the regularization. When μ increases, large stretch factors are penalized and a more smooth result is achieved. Overly smooth results, however, are also undesirable, potentially requiring a user to adjust the parameter as desired.

5.3. Straight-to-Swing Warping

To demonstrate the ability of rhythmic warping with stiffness control, Fig. 8 shows the result of applying Eq. (4) to the situation of Fig. 5. Given a fixed length, the amplitude decay of the four individual notes smoothly stretch or compress to achieve the swing effect, but in a way that preserves note onsets. For exceedingly large overall stretch factors, such control can greatly improve the output sound quality. A similar result using flexure points would require considerable effort and could potentially require updated flexure points for each adjustment of the overall length.

5.4. Rhythmic Emphasis Modification

As an example of rhythmic emphasis modifications, the proposed method is applied to a speech recording of the famous quote "I'm gonna make him an offer he can't refuse," as spoken by actor Marlon Brando in *The Godfather*. The original recording from the movie is spoken at a fairly constant pace. For theatrical effect, we

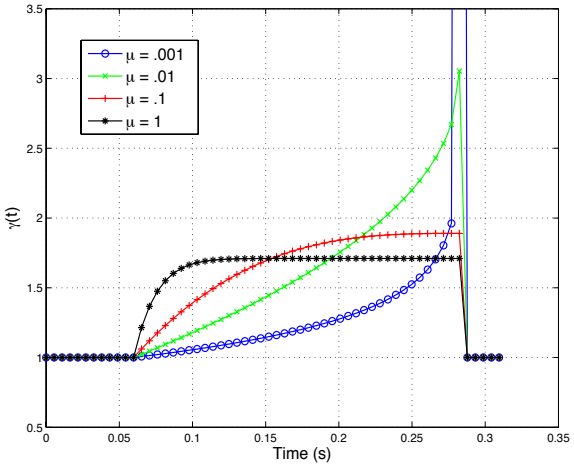

Figure 7: *Results of varying the regularization parameter* μ. *Only one-fourth of the audio signal displayed in Fig. 5 is shown.*

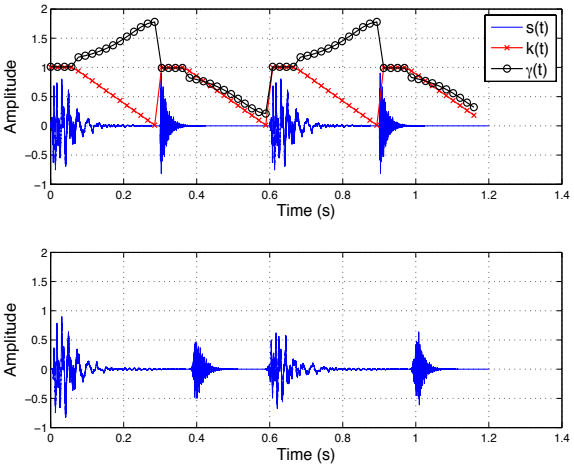

Figure 8: *Straight-to-swing rhythmic warping. The annotated stiffness and computed stretch factor* γ *with input audio (upper). The resulting audio output warped to a swing rhythm (lower).*

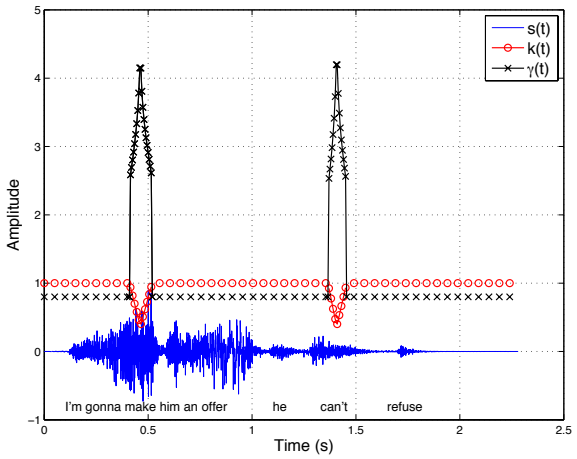

Figure 9: *Rhythmic emphasis modification of* The Godfather's *"I'm gonna make him an offer he can't refuse." The words* make *and* can't *are assigned relatively lower stiffness values.*

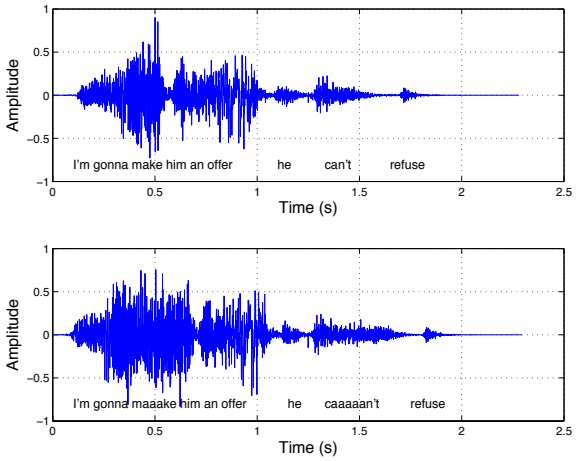

Figure 10: *Input speech recording from* The Godfather *(upper) and output speech with emphasis on the words* make *and* can't *(lower).*

can lengthen the words *make* and *can't* by assigning a lower stiffness to the individual words compared to the rest of the sentence. Then, given an overall output length, the process will stretch or compress the words *make* and *can't* more than other portions of the recordings. Note that the process can maintain the same overall length of the original recording (to maintain video alignment) or can be stretched to a new length.

Figure 9 shows the original recording with annotated stiffness and the resulting stretch factor found by solving the minimization problem with an overall length set to that of the original. Figure 10 then shows the original input recording compared to the output recording. Notice the overall recording lengths are identical, but the amount of time spent on individual syllables within the utterance is warped. One can imagine such effect should prove useful for similar tasks, such as automatic dialogue replacement where actors are required to overdub previously recorded speech that is synchronized to video.

6. CONCLUSIONS

A new method of user control over variable-rate time-stretching is proposed. The method allows a user to independently specify a time-dependent stiffness curve, along with timing constraints, such as an overall length or rhythmic quantization, applicable for time-stretching with transient preservation, rhythmic warping, rhythmic emphasis modification, or similar effects. The user-guided stiffness and timing constraints are then translated into an optimal time-dependent stretch rate using a constrained optimization program. The output of the process yields an optimal time-dependent stretch rate that is then used to generate the desired variable-rate effect using any suitably modified pre-existing time modification algorithm. The metaphor of stiffness is directly motivated by a physical spring chain system and allows a user to specify how each region of sound is stretched relative to one another. Initial results are demonstrated using both a phase-vocoder and pitch-synchronous overlap-add processor, showing promising results.

7. ACKNOWLEDGMENTS

This work was enabled by National Science Foundation Creative IT grant No. IIS-0855758 as well as the funding from the School of Humanities and Sciences, Stanford University.

8. REFERENCES

[1] Tony S. Verma and Teresa H. Y. Meng, "Time scale modification using a sines+transients+noise signal model," in *In Proc. Digital Audio Effects (DAFx)*, 1998.

[2] F. Gouyon, L. Fabig, and J. Bonada, "Rhythmic expressiveness transformations of audio recordings swing modifications," in *In Proc. Digital Audio Effects (DAFx)*, 2003.

[3] Peter Møller Nielson and Steffen Brandorff, "Time-stretching with a time dependent stretch factor," *University of Aarhus*, 2002, `http://www.daimi.au.dk/~pmn/sound/`.

[4] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," Apr. 2011.

[5] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura, Eds., Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, 2008.

[6] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, NY, NY, USA, 2004.

[7] D. P. W. Ellis, "A phase vocoder in Matlab," 2002, `http://www.ee.columbia.edu/~dpwe/resources/matlab/pvoc/`.

[8] P. Dutilleux, G. De Poli, A. von dem Knesebeck, and U. Zoelzer, "Time-segment processing," in *Dafx: Digital Audio Effects*, Udo Zoelzer, Ed. John Wiley & Sons, Inc., NY, NY, USA, second edition, 2011.

[9] Alain de Cheveigné and Hideki Kawahara, "Yin, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am*, vol. 111, no. 4, pp. 1917–1930, April 2002.

[10] J. Laroche and M. Dolson, "Improved phase vocoder time-scale modification of audio," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 323 –332, may 1999.