Target Acquisition vs. Expressive Motion: Dynamic Pitch Warping for Intonation Correction

Olivier Perrotin, Christophe d'Alessandro

Target Acquisition vs. Expressive Motion: Dynamic Pitch Warping for Intonation Correction

OLIVIER PERROTIN, LIMSI, CNRS, Université Paris-Saclay
CHRISTOPHE D'ALESSANDRO, LIMSI, CNRS, Université Paris-Saclay

The purpose of pitch correction is to assist a musician in playing notes with accuracy and precision, without preventing expressive pitch variations. This study presents and examines a new method for automatic pitch correction: Dynamic Pitch Warping (DPW). The analytic formulation of the warping function is derived. In the context of live playing of continuous melodies, i.e., pitch trajectories, the dynamics of pitch correction must be considered. Methods for triggering and releasing the correction are discussed, and a performance test is conducted. DPW is evaluated in the context of digital musical instruments that are controlled by a stylus on a graphic tablet. The results show significant improvement in note accuracy and precision with the addition of the correction method. Analyses of various types of modulations (including vibrato, portamento, and glissando) demonstrate that expressive pitch variations are preserved by the DPW correction. Perceptual tests show that the effects of DPW correction are well perceived and positively assessed by listeners. The proposed method allows for accurate pitch target acquisition together with preservation of expressive motion, a result that could be extended to other situations that require dynamic trajectory correction.

CCS Concepts: •**General and reference** → **Evaluation; Design;** *Performance;* •**Human-centered computing** → **Pointing devices; Pointing; Gestural input;** *Usability testing; Laboratory experiments;* Auditory feedback; •**Applied computing** → **Performing arts;** Sound and music computing;

Additional Key Words and Phrases: Adaptive mapping, dynamic pitch correction, target acquisition, expressive pointing, gestural control of pitch, digital musical instrument

ACM Reference Format:
Olivier Perrotin and Christophe d'Alessandro, 2015. Target Acquisition vs. Expressive Motion: Dynamic Pitch Warping for Intonation Correction. *ACM Trans. Comput.-Hum. Interact.* 0, 0, Article 0 (May 2015), 21 pages.
DOI: 0000001.0000001

1. INTRODUCTION: PITCH ACCURACY AND PITCH EXPRESSIVITY

A Digital Musical Instrument (DMI) is the result of a combination of a human-computer interface (HCI) and a sound generation (synthesis, sampling) engine. The present study focuses on the HCI side of this association. Interactions with a DMI often result in complementary musical tasks, including the production of specific tones with definite pitch, loudness, and timbre and the expressive modulation of these features. As discussed by [Orio et al. 2001], the playing of a specific tone by a musician can be considered a target acquisition task, i.e., targeting a given note of a predefined scale for melodic instruments. Western music, for example, is based on the so-called chromatic scale, which is the division of the octave into 12 equal-size intervals called *semitones* (ST). Pitch acquisition is a highly sensitive task because even a slightly out-of-tune note is noticeable and immediately degrades the performance rendering, regardless of the sound quality. At the same time, expressive musical interpretation

Author's addresses: O. Perrotin and C. d'Alessandro, LIMSI, CNRS, Université Paris-Saclay, Bât. 508, rue John von Neumann, Campus Universitaire, F-91405 Orsay, France

DOI: 0000001.0000001

often requires pitch modulations such as *portamento* or *glissando*, i.e., continuous and smooth (fast and slow, respectively) transitions between notes; or *vibrato*, i.e., pitch oscillation around an exact note's pitch when the instrument allows it. Thus, accurate and precise target acquisition is the most important requirement (for playing in tune), whereas expressive variations are needed for valuable musical performances.

For DMIs that are equipped with a traditional piano-like keyboard, target acquisition is only a key selection task. The pitch played for a given key is always tuned to a predefined musical scale. Although no pitch correction is needed, no expressive pitch modulation is possible either without additional devices. Conversely, playing in tune is more difficult for DMIs that allow for continuous pitch control through devices such as continuous keyboards, graphic tablets, tangible tables, smartphone pads, or other types of continuous surfaces. In the same continuous space, the player has to target accurate notes of a musical scale, but she/he is also able to perform expressive pitch variations.

In acoustic music, continuous pitch control is possible with vocals or fret-less string instruments (e.g., violin family, fret-less guitars), where pitch modulation is an integral part of the instrument's identity. The musician thus faces a double-edged issue: playing in tune (accurate pitch) or playing expressively (keeping full control of pitch). As such, the price for increased expression is increased difficulty: for a novice player, playing a continuous DMI in tune is significantly more difficult than playing a keyed DMI.

Consequently, there is a tradeoff between, on the one hand, helping the user play in tune (for instance, by always outputting the note closest to the current finger position, which limits the possible pitches to the discrete set of a chromatic scale) and, on the other hand, preserving the user's continuous control over the output pitch (to enable effects such as vibrato). Two main approaches coexist to cope with this issue. In the first approach, a discrete scale is used, and expressive modulations around the targets are enabled using additional devices, for instance, a pitch bend wheel (the Minimoog or Yamaha DX7) or an expression pedal. In the second approach, the problem is addressed from the opposite direction: a continuous pitch control device allows for expressive modulations, and discrete notes are targeted using a pitch correction algorithm. This second approach is developed and evaluated in the present work using a new algorithm called *Dynamic Pitch Warping* (DPW). The method is inspired by target acquisition methods in the visual-spatial domain but possesses new specific features for preserving expressive variations. This situation is also encountered in other HCI tasks, where acquiring a target is the primary goal but where variations in the manner the target is acquired or reached are also of interest. The present algorithm could hopefully be extended to other such applications that require dynamic trajectory correction. The following sections present previous work (section 2), our method (section 3), an evaluation (section 4) and discussion (section 5).

2. A REVIEW OF POINTING AND PITCH ENHANCEMENT METHODS

2.1. Pointing task enhancement in the visual domain

Current methods for pitch correction are inspired from the visual domain correction methods that are used for cursor pointing enhancement. Playing musical notes on a musical interface in the audio domain and reaching icons on a screen with a cursor in the visual domain are comparable spatial pointing tasks in the motor domain, though they differ in their perceptual control modality. The user's performance in both types of tasks can be enhanced with the help of a correction algorithm.

2.1.1. Pointing correction in GUI systems. Two main methods to help users reach targets have been investigated in the context of the mouse interface for pointing tasks using a Graphical User Interface (GUI): enlarging the target width and reducing the amplitude of movement [Balakrishnan 2004]. Visually expanding targets when the cursor is nearby using the *Expanding targets* method of [McGuffin and Balakrishnan 2005] by adding a bubble around them [Cockburn and Firth 2003] or by extending the selection area of the cursor [Chapuis et al. 2009] aids target acquisition. It appears that users learn to anticipate the expansion and aim at the area around the target, not at the target itself. However, these methods introduce distortions in the visual display, thus potentially altering the acquisition of adjacent targets.

To expand targets without modifying the visual display, one solution is to enlarge them in the motor domain. The control-display ratio (C-D ratio) is a gain that maps the user's movement in the motor domain (in physical distance) to the corresponding displacement in the visual domain (in pixels) [MacKenzie and Riddersma 1994]. Decreasing the C-D ratio locally around a target makes the latter larger in the motor domain and, therefore, easier to reach [Blanch et al. 2004]. This is called the *Sticky Icons* method [Worden et al. 1997]. Conversely, increasing the C-D ratio during displacement towards the target reduces movement amplitude in the motor domain.

Comparisons between the *Expanding targets* and *Sticky icons* methods for target acquisition show similar performance [Cockburn and Firth 2003]. Nevertheless, the *Sticky Icons* method is often preferred because it is less intrusive and does not introduce visual distractions that are difficult to handle when numerous potential targets are displayed at the same time.

2.1.2. Distant pointing correction. Target acquisition appears less precise for distant pointing techniques than it is with the mouse [Polacek et al. 2012], [Pino et al. 2013]. Examples of distant pointing techniques are remote pointing (e.g., the Nintendo Wii), free-hand pointing (e.g., the Microsoft Kinect), or hybrid techniques using a stylus to point at distant targets on a tabletop [Parker et al. 2005]. Methods for pointing correction have been adapted for both stylus distant pointing [Parker et al. 2005] and free-hand pointing [Mäkelä et al. 2014]. In general, the *Sticky icons* method offers better performance than does the *Expanding targets* method for these applications.

2.2. Pitch enhancement in the audio domain

Real-time musical pitch adjustment methods appeared approximately two decades ago for improving vocal performances. Two main steps are involved. First, a pitch detection algorithm estimates the singers' pitch. Second, the vocal sound is post-processed by a pitch-scaling algorithm to obtain the correct pitch. Auto-tune[1] [Hildebrand 1999] is a widely used voice pitch correction system. The vocal pitch discretization results in audible artifacts, and this type of distortion effect is both often sought after and highly prized in popular singing. In contrast, for a DMI, pitch correction must remain unnoticeable with respect to sound quality, and sound distortion must be avoided altogether. Compared to singing, the process is simpler because the first step, pitch detection, is not required: the pitch is given directly by the interface.

For improving usability, several continuous-pitch DMI instruments are equipped with some sort of pitch correction algorithm. Examples of such instruments are the *Continuum Fingerboard*[2] [Haken et al. 1998], the *Seaboard*[3] [Lamb and Robertson

[1] http://www.antarestech.com/products/detail.php?product=Auto-Tune_7_1 last checked: October, 19th 2015.
[2] http://www.hakenaudio.com/Continuum/ last checked: October, 19th 2015.
[3] https://www.roli.com/ last checked: October, 19th 2015.

Fig. 1. Examples of warping functions. From left to right: continuous; discrete; fixed; adaptive functions.

2011], the *LinnStrument*[4], the iOS applications *MorphWiz*[5] and *Garageband*[6], the *TouchKeys* [McPherson et al. 2013] and *Cantor Digitalis*[7] [Perrotin and d'Alessandro 2013]. For each of these examples, pitch is continuously controlled by the position of a hand-driven mobile object (e.g., finger or stylus) on a continuous surface. The mobile object's position on the surface defines the *input pitch*, which is processed and ultimately results in a played pitch, or *output pitch*. These DMIs use three main methods to map input to output pitch: magnet correction, warping correction and dynamic warping correction.

2.2.1. Magnet correction. In magnet correction, the output pitch is continually attracted toward the closest note of a predefined musical scale by changing the value of the output pitch incrementally. The first and simplest magnet correction method is Auto-Tune [Hildebrand 1999]. This is a pitch convergence method with one degree of freedom: the "retune speed". This parameter sets the duration of the output pitch adjustment. A retune speed of 0 ms is equivalent to a keyboard-like mapping correction. In contrast, longer retune speeds allow for some expressive pitch variation[8].

The pitch correction in the *Continuum Fingerboard* [Haken 2009] also uses a magnet correction method. The convergence is set by two parameters: a correction step, CS, and a time step, TS. At regular time intervals, the correction step is added to or subtracted from the input pitch such that the general trend of the pitch gets closer to the target. The two degrees of freedom are the correction frequency set by the time step, and the rate of expression preservation set by the ratio CS/TS. Every pitch modulation whose frequency is higher than $1/TS$ and whose velocity is lower than CS/TS will be distorted. Therefore, this ratio must be lower than the velocity of any pitch modulation to be preserved. There is a fixed trade-off between accuracy and expression preservation in the magnet correction method.

2.2.2. Warping correction. In warping correction, output pitch is computed as a time-invariant *warping function* of input pitch, as illustrated in Figure 1. An output pitch identical to the input is obtained with the linear function (first panel) that is typical of continuous instruments. Another example is the staircase function (second panel), which transforms an input pitch continuum into a discrete-steps output pitch, similar to a keyboard. No expressive pitch modulation is possible because pitch transitions between notes are always abrupt. Smoothed staircase functions (third panel) are often chosen to compromise between accuracy and loss of expressivity [Goudard et al. 2014]. Smoothed staircase functions are similar to those used for the visual *Sticky Icons* correction method because the gain between input and output pitches is reduced around the targets. The last panel shows an example of an adaptive function, which is computed from the input pitch value. The principle is derived from the visual *Expanding targets* correction; the warping function matches the target for any input position. The adaptive function was first introduced to correct the pitch of note attacks [Perrotin and d'Alessandro 2013]. This method is presented in more detail below.

[4]http://www.rogerlinndesign.com/linnstrument.html last checked: October, 19th 2015.
[5]http://www.wizdommusic.com/products/morphwiz.html last checked: October, 19th 2015.
[6]https://www.apple.com/ios/garageband/ last checked: October, 19th 2015.
[7]http://cantordigitalis.limsi.fr/ last checked: October, 19th 2015.
[8]This product includes other options, such as the choice of a second "retune speed" for sustained notes, the addition of an artificial vibrato, and the expansion of natural vibrato.

Fig. 2. DPW correction example. Input and output pitches are the thick grey dotted and solid black lines, respectively. The horizontal solid lines are exact notes. The horizontal thin dotted lines delimit detection intervals $I = 0.1$ ST. Four zones are highlighted: (1) The input pitch remains in a detection interval I longer than the critical time $T_c = 100$ ms; (2) the correction is applied during the transition time T_t = 50 ms; (3) the output pitch evolves with the warping function; (4) a linear function is applied after the next semitone is reached (F#).

2.2.3. Dynamic warping correction. One interesting way to add flexibility to warping correction methods is to compute the warping function depending on the mobile object's dynamics. The *TouchKeys* [McPherson et al. 2013] instrument is equipped with a two-degree-of-freedom dynamic warping correction. The controller is an augmented piano keyboard that allows players to bend pitches by sliding their fingers lengthwise on depressed keys. The two parameters to this operation are the "snap zone" size around the targets and a speed threshold. The correction is applied when the pitch enters in a snap zone with a speed lower than the threshold. The output pitch adjustment function and the correction triggering parameters, such as the pitch speed threshold, were not explicitly provided in the publication.

Dynamic warping correction introduces a speed threshold, thus enabling a distinction between stable notes (that must be corrected) and pitch modulations (that must not be corrected). This distinction is based on the rate of pitch change (pitch velocity) because pitch expressive modulations are likely to vary faster than stable notes. Pitch is corrected only when the input pitch velocity is lower than some critical pitch velocity threshold. To go beyond correction of pitch bends, a method with three-degrees-of-freedom method, i.e., the Dynamic Pitch Warping correction, is proposed in the next section.

3. DYNAMIC PITCH WARPING CORRECTION

With Dynamic Pitch Warping (DPW), the output pitch depends on the dynamics of the input pitch. The goal of the proposed correction method is to map the output pitch to the target note only when the input pitch is stable enough to assume that the player's intent is to target a stable note rather than an expressive modulation. Following a brief description of the principle behind Dynamic Pitch Warping, the pitch warping function is discussed. Then, the correction dynamics and parameters are described. Examples of different dynamics are explored at the end of this section.

3.1. Dynamic Pitch Warping principle

An example of DPW correction is shown in Figure 2. This example shows an input pitch (thick dark grey dotted line) and output pitch (thick solid black line) for the melody F-G-F played by a musician. The thin dotted lines subdivide the pitch axis in small and equal intervals called detection intervals I. The four labeled zones correspond to the 4 steps of the correction process:

(1) The input pitch enters a detection interval, I, and remains within the interval for more than the critical time, T_c. At that moment, the algorithm considers the input pitch trajectory sufficiently stable to be corrected. Correction is triggered (right border of zone 1).
(2) The correction is smoothly applied during transition time, T_t, attracting the output pitch to the target note represented by the plain horizontal lines.
(3) The output pitch is computed from the pitch warping function and converges to the input pitch when the next exact note is reached (F#).
(4) A linear function is applied after the next exact note is reached (F#) so that the curves overlap until a new correction is applied.

Fig. 3. Pitch warping function: output pitch as a function of input pitch. Here, (0,0) corresponds to the targeted exact note, while $(-\delta,-\delta)$ and (δ,δ) are the positions of the previous and next exact notes respectively. Left: example of input pitch at x_{01} at the instant of correction; Right: input pitch at x_{02} at the instant of correction.

The three degrees of freedom of this method are the detection interval size, the critical time, and the transition time. The first two parameters are equivalent to the "snap zone" and "speed threshold" parameters, whereas the third additional parameter–the transition time–is similar to the "retune speed" parameter. These parameters are set *a priori* by the musician and are discussed in section 3.3. In summary, the correction is triggered only when the input pitch is stable for a substantial period of time. When that occurs, the correction is triggered, and the pitch is corrected such that the note played is in tune for the current input pitch because the mapping dynamically adapts to the current position. If the input pitch is moving in any particular direction, the pitch changes according to the warping function, thus allowing for pitch modulations.

3.2. Pitch warping function

Consider a pitch scale with equidistant intervals of size δ (where each exact note is a multiple of δ). For the sake of simplicity, the warping function is expressed relative to the currently targeted note, i.e., the closest note in the scale. The latter is given the coordinates (0,0) in the (input pitch $\times$ output pitch) plane. $(-\delta,-\delta)$ and (δ,δ) correspond to the previous and next exact notes in the scale, respectively. The warping function is applied to the interval [-δ,δ] around the targeted pitch (see Figure 3). The example in section 3.1 uses the semitone (ST) scale, where $\delta = 1$ ST.

The warping function must fulfill two conditions:

— When corrected, the current input pitch x_0 must correspond to the closest note in the scale for the output pitch. Then, the warping function is 0 for x_0 at the instant of correction to the correct pitch. Specific warping functions are computed depending on the input pitch (see x_{01} and x_{02} in Figure 3).

— To avoid a persistent shift between the input pitch and output pitch, the warping function must converge to the linear pitch at the previous and next exact notes of the scale, i.e., the warping function must go through the points $(-\delta,-\delta)$ and (δ,δ).

To meet these constraints, the warping function is an arch-shaped function with curvature γ depending on the initial input pitch x_0. A simple form of an arched function with a given curvature can be defined using an exponential function. Because it is bijective, the inverse function is used for the sake of simplicity. The output pitch y as a function of the input pitch x is given by

$$\begin{cases} y(x) \;=\; \frac{1}{\gamma}\left[\log\left[(e^{2\gamma\delta}-1)(\frac{x}{\delta}+1)\frac{1}{2}+1\right]\right]-\delta & \text{if } \gamma \neq 0 \\ y(x) \;=\; x & \text{if } \gamma = 0 \end{cases} \tag{1}$$

A new warping function, i.e., a curvature γ, is computed each time the correction is triggered. The condition for the exact pitch at the initial position x_0 is $y(x_0,\gamma_0) \;=\; 0$, where γ_0 is the curvature of the function at the instant the correction is triggered. This leads to the following expression:

$$\gamma_0 = \frac{1}{\delta}\log\left(\frac{\delta - x_0}{\delta + x_0}\right) \tag{2}$$

These equations (1 and 2) are valid for any regular musical scale. In the next sections, the semitone scale is used, where δ = 1 following the MIDI (Musical Instrument Digital Interface) convention.

3.3. Correction dynamics

Application of the warping function from section 3.2 for all notes at any time would result in discrete pitch steps. Therefore, additional degrees of freedom are introduced to apply the correction as a function of the input pitch dynamics. The correction dynamic depends on the way in which a note is played and falls into one of two categories. In detached or *staccato* notes, each note of a melody corresponds to a new contact between the mobile object and the interface; there is a clear release at the end of each note. In *legato* notes, the mobile object (such as a finger or stylus) remains in contact with the surface without a clear release between successive notes.

For *staccato* notes, the correction is activated at the time at which the stylus hits a graphic tablet or a finger hits a continuous surface to ensure that the note produced is in tune at the mobile object's first contact. This is called *Onset Correction* or OC. Note that activation of OC prevents initial ornaments that may be required; for example, it impedes note attacks "from above" or "from below" that are typical of some musical styles.

In the case of *legato* notes, the pitch contour must be continuously adjusted to meet the notes of a musical scale without distorting pitch modulations. This is called *Contour Correction* or CC. In this case, the correction needs to be applied only for stable notes, as explained above and illustrated in Figure 2. In addition, correction is applied and released smoothly to avoid jumps in the output pitch at the instant of correction. The parameters controlling correction triggering, correction application, and correction release, i.e., the three degrees of freedom of the DPW correction method, are described in this section.

3.3.1. Triggering the correction. When playing *legato*, i.e., when the input pitch is moving continuously from one note to another, the resulting contour contains both the targeted notes and any expressive pitch modulations. Because the dynamic range of expressive pitch modulations is larger than the dynamic range of targeted notes, the input pitch velocity is useful for distinguishing unintended pitch modulation around targeted notes from intended expressive pitch modulations. The correction could be triggered when the input pitch's instantaneous velocity falls below a given threshold; however, this would prevent expressive modulations such as vibrato; every change in pitch direction (null instantaneous velocity) would trigger correction. Thus, some kind of average input pitch velocity must be considered instead of the instantaneous velocity.

First, we define the input pitch critical average velocity, AV_c, under which the pitch motion is considered stable–the conditions under which correction can be applied. The critical average velocity is expressed as $AV_c = I/T_c$, where I is the detection interval and T_c is the critical time. When the input pitch enters the interval I, a timer is started. If the timer exceeds T_c before the input pitch leaves the interval, the input pitch average velocity is lower than AV_c (zone (1) of Figure 2). This indicates a steady note; therefore, correction is subsequently applied. Conversely, if the input pitch leaves the interval before the timer reaches T_c, the input pitch average velocity is higher than AV_c, and no correction is applied.

Detection intervals I are obtained by dividing the input pitch axis into a number of equal intervals per δ, as shown in Figure 2 by the horizontal dotted lines. The size of the detection interval defines the pitch correction granularity. For small intervals (e.g., 0.1 ST), the input pitch is likely to change frequently between intervals, each time requiring a check for a stable trajectory (and possibly triggering the correction). Consequently, small intervals improve pitch accuracy, but they may limit expressive

ALGORITHM 1: Trigger the DPW correction

Input: Input pitch, I, T_c
Output: Triggering signal

```
// Trigger the DPW correction (zone 1)
while the DPW correction is on do
    if a contact is made by the mobile object and OC mode is on then
        Send triggering signal;
    end
    while the mobile object remains in contact with the interface and CC mode is on do
        Set timer T = 0;
        while the input pitch remains in a detection interval I do
            Increment timer T;
            if T > T_c then
                Send triggering signal
            end
        end
    end
end
```

modulations. Larger intervals (e.g., 0.5 ST) allow for more expression preservation, but unintentional deviations within the intervals will not be detected.

After determining a detection interval, the critical time T_c parameter defines the critical average velocity AV_c. This value acts as the threshold between correction reactiveness and expression preservation. A relatively small vibrato with $\pm$ 0.1 ST extent and a 5 Hz rate (as reported in choir singing [Sundberg 1994]) gives an average velocity of approximately AV = 2 ST/s. A critical average velocity lower than 2 ST/s should therefore preserve natural vocal vibrato. However, one can find glissandos that evolve more slowly. A smaller average velocity (a longer critical time) can be considered to preserve glissandos–but only at the expense of reactiveness.

3.3.2. Applying the correction. For the first note of a melody, OC is triggered, and the pitch is adjusted immediately such that the note is in tune independently of the position of the initial input pitch. For subsequent notes, CC pitch correction is smoothly applied to avoid audible pitch steps or jumps. A transition time T_t is employed to achieve smooth pitch transitions, thereby minimizing the perception of correction. The longer the transition time is, the less perceptible the correction is, but the more audible pitch inaccuracies are. Pitch correction is applied by continuously changing the curvature γ of the pitch warping function from its initial shape to the corrected shape with curvature γ_0 (equation 2) during the transition time T_t. If the input pitch evolves during the correction transition, the output pitch is modified according to the changing warping function. If a new correction is triggered before the end of the previous correction transition, a new γ_0 is computed, and the curvature changes accordingly.

The transition time between two defined notes for trained and untrained singers seems almost constant between intervals [Sundberg 1973], [Xu and Sun 2000], with a duration of about 140 ms. The response time of trained choir singers for pitch adaptation is also on the order of approximately 140 ms, discounting singers' reaction time [Grell et al. 2009]. Therefore, 140 ms serves as a reference for the transition time. A longer correction transition time (e.g., 250 ms) could be used if the goal is a smoother correction, with almost unnoticeable pitch transitions. In contrast, a shorter correction transition time (e.g., 50 ms), although audible, allows for faster pitch correction. The correction transition time is exemplified in Figure 2, represented in zone (2).

ALGORITHM 2: Compute the warping function

Input: Input pitch, T_t, Triggering signal
Output: Current target, Warping function

```
// Apply the correction (zone 2)
```

if *the triggering signal is received* **then**
  Lock the target to its current value;
  Given the current input pitch position x_0, compute the curvature γ_0 with equation 2 to make the user play in tune;
  if *the triggering ensues from a new contact* **then**
    Set the current warping function with equation 1, with $\gamma = \gamma_0$;
  else
    Set the current warping function with equation 1, continuously changing the current γ to γ_0 in time T_t;
  end
end

```
// Release the correction (zone 4)
```

if *input pitch leaves the interval* $[-\delta, \delta]$ *around the target position* **then**
  Compute the target as the closest exact note from the input pitch;
  Immediately Set the current warping function to a linear function (γ = 0);
end

3.3.3. Releasing the correction. Pitch correction stops when the input pitch leaves the interval $[-\delta, \delta]$, in other words, when it reaches the next note on the musical scale. At this point, a linear function is applied between the input and output pitches. The function change is not audible because the warping function converges to a linear function at the extremities of $[-\delta, \delta]$. This is represented at the border between zones (3) and (4) in Figure 2.

3.3.4. Final algorithms. The DPW correction is summarized by algorithms 1 (triggering the correction), 2 (warping function computation), and 3 (pitch correction).

It should be noted that the three parameters of DPW (namely, detection interval I, critical time T_c, and transition time T_t) subsume the parameters of the two magnet corrections presented in section 2.2.1, thereby controlling the correction granularity, the rate of expression preservation, and the rate of correction convergence (linked to the correction audibility), respectively.

3.4. Preservation of expressive modulations

Parameters I and T_c play an essential role in preserving pitch expression. To illustrate the role of these parameters, several examples of parameter settings are presented here, and the output pitch is analyzed for typical expressive input pitch patterns, including the main pitch modulations encountered in Western music. The average velocity AV_c in DPW correction and the convergence speed CS/TS in the Haken magnet

ALGORITHM 3: Pitch correction

Input: Input pitch, Current target, Warping function
Output: Output pitch

Compute the input pitch relative to the current target;
Compute the output pitch from the relative input pitch with the current warping function;

Table I. Triggering settings used in expressivity analysis.

	DPW Correction		
	Detection Interval I	Critical Time T_c	Critical Average Velocity AV_c
A	0.1 ST	0.1 s	1 ST/s
B	0.5 ST	0.25 s	2 ST/s
C	0.1 ST	0.25 s	0.4 ST/s
	Haken Correction		
	Correction Step CS	Time Step TS	Correction Slope CS/TS
D	0.01 ST	0.005 s	2 ST/s
E	0.01 ST	0.025 s	0.4 ST/s
F	0.001 ST	0.005 s	0.2 ST/s

Fig. 4. Examples of corrections (left: DPW correction, right: Haken correction) with different settings (see table I). The thin dark dashed lines are the input pitches (stylus trajectories); the thick plain black lines are the output (played) pitches; and the horizontal plain lines are the target notes. The horizontal thin dotted lines represent the intervals of detection I (B: 0.5 ST; A and C: 0.1 ST). The darkest to the lightest backgrounds represent zones 1 to 4 of the DPW correction, respectively. The corresponding audio and video examples are referenced in Table III.

correction have the same purpose: expression preservation. Expressive variations are preserved only if AV_c or CS/TS is lower than the variations' average velocity. The DPW and Haken corrections are compared using the parameter settings presented in Table I.

Settings A to C are used for DPW correction. Setting A presents a small detection interval and a small critical time, giving a reactive correction. Setting B proposes a large detection interval and a long critical time, leading to fewer corrections. Setting C is in between, specifying a smaller critical average velocity. The transition time T_t = 50 ms is used for all three settings. Settings D to F are used for the Haken correction. Setting D is the same as the example in [Haken 2009]. The slope of correction CS/TS is decreased by increasing TS (setting E) or by decreasing CS (setting F).

These different methods and settings were applied to a melodic pattern exhibiting the three types of modulation mentioned above (portamento, glissando and vibrato). The input pitch contour is organized as follows: start with the note C, followed by two successive portamentos up to the notes C# and D, add vibrato and then a glissando back down to the note C. The input and output pitches are shown in Figure 4 with thick dashed and solid lines, respectively. The horizontal thin lines are the pitch targets, and the dashed horizontal lines are the limits of the detection intervals. The same background colors are used as in Figure 2: the darkest to the lightest colors represent zones 1 to 4 of the DPW correction, respectively. Panels A to F correspond to the settings of Table I. To help understand the method, audio and video examples of the different corrections and settings are also available. These examples are summarized in Table III.

Panel A shows the effect of fast DPW correction, i.e., a short detection interval I with a short critical time T_c. Because of these small values, the correction is triggered often and attempts to adjust the pitch during the beginning and end of the vibrato, leading to small distortions. This happens when the vibrato amplitude is close to the detection interval size. With a critical time smaller than the vibrato period (150-200 ms), the input pitch is likely to remain long enough in the detection interval to trigger the correction within a vibrato cycle. This setting controls the limit of vibrato preservation. One could expect even more distortions with a smaller critical time. Along the same line, the correction is maintained during the glissando. Because the pitch is constantly corrected, the result is that the pitches occur in steps rather than sliding from one to

Fig. 5. Visual cues applied to the tablet surface. Pitch is controlled by stylus position on the x-axis. The equidistant vertical lines correspond to a semitone scale.

another. In summary, a too-fast correction becomes intrusive and prevents some forms of expressive pitch motion.

The effects of slow DPW correction (a large detection interval and a long critical time) are shown in panel B. Portamentos are well corrected, as is the vibrato; the correction applied after the second portamento is maintained throughout the vibrato duration. Only one correction is triggered during the vibrato, due to a change of detection interval, but it introduces no distortion. During the glissando, although the corrections happen less frequently than in setting A, the critical average velocity is twice as high, so the pitch is still corrected and mapped to the closest exact notes, which results in undesirable peaks in the trajectory.

The two previous cases indicate that for effective modulation preservation, the interval I needs to be small enough for a reactive correction, but the critical time T_c must be sufficiently long to preserve the modulations. Panel C shows the correction applied with $I = 0.1$ ST and $T_c = 250$ ms, allowing a critical average velocity of 0.4 ST/s, which is slower than some glissandos. The results are globally similar to those of setting B for portamento and vibrato. However, because the critical intervals are smaller, the input pitch does not remain inside them long enough to trigger the correction during the glissando. Therefore, no distortions are introduced for any of the three effects with this setting.

With the Haken correction, every variation of the input pitch lower than the correction slope CS/TS appears to be attracted more abruptly to the closest exact note. In panel D, the correction slope is high (CS/TS = 2 ST/s). As a result, the glissando is transformed into steps–even more severely than in panel A. The vibrato is also distorted. In panel E, the slope CS/TS = 0.4ST/s is reduced to match the critical average velocity of panel C. Although the steps are no longer present in the glissando correction, a small distortion is still present in the vibrato. The slope CS/TS is again reduced to 0.2 ST/s in panel F; however, a small vibrato distortion remains, and the slope has become too low to correct the portamentos as successfully as in panels B or C.

Comparisons between panels D, E, and F show that the magnet correction provides a choice between accuracy (fast correction, steep correction slope) and expressivity (slow correction, shallow correction slope) but fails to manage both constraints simultaneously. In contrast, comparisons between panels C, E, and F show that the DPW correction is able to manage both accuracy and expressivity at the same time. It should be noted that the tempo is relatively slow in this example. A faster tempo would require a shorter critical time to correct short notes at the expense of the loss of some expressive modulations. In summary, the DPW method enables expressive modulation preservation, introducing minimal distortions for the three effects when appropriate settings are used.

4. DYNAMIC PITCH WARPING EVALUATION

DPW performance has been assessed using two evaluation types. The first experiment examines the objective pitch accuracy and precision of the output pitch. The second experiment perceptually evaluates the effect of various pitch correction conditions.

4.1. DMI used for the two experiments: Cantor Digitalis

DPW correction was initially developed in the context of the *Cantor Digitalis*, a real-time singing voice synthesizer controlled by a graphic tablet [Feugère 2013]. This DMI was used in both the objective and subjective experiments. *Cantor Digitalis*' synthesis

Fig. 6. Melodic patterns used for the experiment.

engine is an improved parametric voice synthesizer. Vocal parameters are controlled in real time by means of a graphic tablet (Wacom Intuos) using both a stylus and fingers. For this study, the control parameter variables were limited to pitch only. The input pitch was determined by the stylus position along the x-axis. The y-axis was unconstrained and not interpreted, allowing more natural and comfortable gestures by the users.

To help the user target the appropriate pitches, visual cues were printed and precisely fixed to the working surface of the tablet, shown in Figure 5. The patterns used depend on the music played (e.g., modal scales for Indian raga, keyboards, guitar neck) In this example, a piano-like keyboard is represented, whose keys (grey and white rectangles) are all the same length. The keys are displayed as an indicator of the closest pitch on the scale, and the actual pitch scale is continuous. The user must target the vertical lines corresponding to the keys' centers to play in tune. If the stylus points at another position, the corresponding in-between pitch is played. An equally-spaced scale was used as a default value (A4=440Hz). The names of the natural notes are indicated. In this example, the mapping corresponds to 6 mm per semitone, with 35 semitones available (from G# to G). The spatial resolution of the stylus position is 0.005 mm, which equates to a pitch resolution of 0.08 cents, far below the difference limens for human pitch perception (which is approximately 1 cent at 440 Hz [Moore 1973]). This interface was chosen due to its high spatial precision (the stylus tip is much thinner than a finger). As a result, the pitch accuracy and precision target task is more challenging than for coarser interfaces.

4.2. Experiment 1: Note accuracy and precision

4.2.1. Protocol. Melodic imitation paradigms are common for the evaluation of accuracy and precision [Dalla Bella et al. 2007], [Pfordresher et al. 2010], [Larrouy-Maestri et al. 2013]. For each stimulus, participants were asked to reproduce a melody with the stylus on the graphic tablet. Four different melodic patterns were presented. The resulting output pitch was recorded both with and without DPW correction. For each stimulus, subjects could listen to the melody and read a visual reference score that included both musical notation and the notes' names. Reference sounds were produced by a MIDI synthesizer (piano sound, equal temperament, A4 = 440 Hz), which could be replayed on command. The performance sounds were generated by the *Cantor Digitalis* synthesizer. Melodic patterns were presented randomly through headphones at a comfortable listening level. Subjects were asked to record 3 trials for each stimulus. Audio and visual metronomes indicated the tempo to follow. Because only contour correction was studied in this experiment, subjects were asked to keep the stylus in continuous contact with the surface during each trial. Participants were not informed as to whether contour correction was activated.

Three corrections were tested: "fast" correction (I = 0.1 ST; T_c = 100 ms; T_t = 50 ms), "slow" correction (I = 0.5 ST; T_c = 250 ms; T_t = 50 ms), and no correction. Values of I and T_c corresponded to settings A and B from Table I. The melodies were 9-note patterns inspired by vocal warm-up exercises. They presented various levels of difficulty, including both increasing and decreasing intervals of various sizes, from seconds to octaves. The melodic patterns are shown in Figure 6. Subjects were asked to play each pattern at two different tempi with each correction setting: 120 beats per minute (b.p.m.), and 240 b.p.m. In summary, each subject was asked to perform 3 repetitions of 4 melodies under 3 different correction settings for each of 2 tempi.

Fig. 7. Accuracy (left column) and precision (right column) expressed in semitone cents for different groups of subjects considering all notes (top) and fine errors only (bottom). For each condition, two boxes are displayed: input values (left) and output values (right).

Ten subjects (2 females, average age 28.8 years and average musical training 12.4 years) participated in the experiment on a voluntary basis. They were divided into three groups: "Non-Musicians" (three subjects with no musical experience); "Musicians" (four subjects with several years of musical practice, average of 10 years); and "*Cantor Digitalis* Players" (three subjects with more than 10 years of musical practice and regular *Cantor Digitalis* players (∼50h of practice)). None of the subjects reported auditory impairments and all were right-handed. A Wacom Intuos 5M tablet equipped with the printed pattern represented in Figure 5 and the corresponding stylus were used for the experiment. Before beginning the experiment, all participants were given prior instruction concerning the tasks and a training session that presented the same stimuli and the same protocol. Each subject required approximately 30 min on average to complete the test.

For each trial, the input pitch corresponding to the trajectory of the stylus on the tablet and the output pitch actually perceived by the player were recorded. Notes were identified from these recordings, detecting the transitions between notes as peaks in the pitch derivative. For each note, input and output values were extracted following the same stylization process as in [d'Alessandro et al. 2014].

Performance analysis was based on pitch accuracy and precision of the played notes; these functioned as statistical measures for the evaluation of the melodic performances [Pfordresher et al. 2010]. For each input or output value, the error was computed as the difference between the recorded value and the targeted note. Pitch accuracy was calculated as the mean of these errors for a set of notes. Precision was computed as dispersion: the standard deviation of the errors for a set of notes. Accuracy and precision scores closer to 0 indicate more accurate and precise playing, respectively.

4.2.2. Fine and gross errors and musical expertise. The correction algorithm adjusts the input pitch to the closest exact note. Consequently, the correction is efficient only if the input pitch is closer to the target note than to any other note (i.e., below a threshold of 0.5 ST). Otherwise, the correction will adjust the pitch to the wrong note, thereby amplifying the initial error. This type of error is termed "Gross Error" (GE) because it is larger than half a semi-tone (pitch is adjusted to the wrong note). Other errors are labeled as "Fine Errors" (FE).

Figure 7 shows the accuracy (left) and precision (right) of subjects in two different categories: notes with GE and FE (top) and notes with FE only (bottom). The three groups of subjects are separated for each category. The boxes contain 50% of the values, and the lines represent the medians.

The statistical significance of accuracy and precision differences among the different groups were calculated using a Wilcoxon rank-sum test by pairs. No significant improvements were observed from input to output values considering all notes, (with FE and GE). However, the less experienced the subject, the less precise the results. A marginally significant degradation of output value precision was observed between the *Cantor Digitalis* Players and Non-Musician groups $(W = 62, p = 0.06)$. For FE (bottom), the correction significantly improved output values precisions for *Cantor Digitalis* Players $(W = 81, p < 0.01)$ and Musicians $(W = 135, p < 0.01)$. The output value precision of Non-Musicians was not improved significantly by correction.

Fig. 8. Accuracy (left column) and precision (right column) expressed in semitone cents for different conditions. Top: correction sets (no correction; fast correction; slow correction). Bottom: tempo sets (120 b.p.m.; 240 b.p.m.). For each condition, input values (left box) and output values (right box) are shown.

In summary, the correction algorithm was efficient only for users who already performed relatively well. For all users, the algorithm was unable to correct gross errors: when an input pitch was too far from the target note, it was attracted to the closest wrong note. Furthermore, the correction required a certain level of pitch stability to work satisfactorily.

4.2.3. Effect of tempo. It is important to examine the performance of DPW for different tempi and the influence of correction reactiveness. Following the results above, the influence of tempo and correction speed are studied using only analysis of the fine errors. Accuracy and precision were computed on different sets of notes: a "Correction" set (containing all the notes played with each correction), and a "Tempo" set (containing all notes played with a similar tempo). In this experiment, the performances of all subjects (Musicians, Non-Musicians, and *Cantor Digitalis* Players) were analyzed together. The Correction set contained three factors: subject factor (10 levels), trial factor (3 levels), and correction factor (3 levels: no correction, fast correction, and slow correction), resulting in 90 measures for accuracy and precision. The Tempo set also contained three factors: the subject factor (10 levels), the trial factor (3 levels), and the tempo factor (2 levels), resulting in 60 measures for accuracy and precision. In addition, the Tempo set contains only stimuli with correction.

The results are reported in Figure 8. Each box presents the three measures of accuracy (left column) or precision (right column) for the 10 subjects. The first row represents the accuracy and precision of the correction group: no correction, fast correction, or slow correction. The effect of tempo is shown in the second row, providing the effect at 120 b.p.m. and the effect at 240 b.p.m..

The overall results show accuracy medians close to 0 in all cases, with dispersion lower than 5 cents. These results are in agreement with those obtained in a previous study examining the gestural control of singing synthesis [d'Alessandro et al. 2014]. The average precision for input values is around 17 cents and has a relatively small dispersion, which on the graphic tablet corresponds to approximately 1 mm, i.e., roughly the width of the stylus point. This is close to the limits of precision available for this task. The corrected values are all under 17 cents, with some being under the limens for pitch perception, and have an extremely small dispersion.

The first row in Figure 8 shows the effect of pitch correction. When pitch correction is active, the dispersion of accuracy is not significantly reduced for the output values. However, the precision of output values is significantly lower than the input values for fast correction ($W = 805, p < 0.01$) and slow correction ($W = 664, p < 0.01$). The DPW method was able to improve subjects' precision by a factor of 2 (i.e., a decrease in precision error). The accuracy and precision of the output values are not significantly different between the fast and slow tempo corrections. The correction performed equally well for both settings. Furthermore, the accuracy and precision of input values with or without correction did not differ significantly. These results suggest that the correction procedure did not influence the participants.

The second row in Figure 8 shows the effect of tempo. The effect is ot noticeable for accuracy but is significant for precision because the median of the output values was 5 cents higher when the tempo was doubled ($W = 256, p < 0.01$). This might be due to a

Table II. Percentage of corrected notes among each note set

	Fast correction	Slow correction	Total
120 b.p.m.	87	66	76
240 b.p.m.	42	11	26
Total	64	38	51

Fig. 9. A comparison of the present study's results with results obtained in [Perrotin and d'Alessandro 2013]. Accuracy (left) and precision (right) of the subjects are expressed in semitone cents in four different conditions: output values both with and without CC (this experiment), and output values with and without OC (previous work).

less stable trajectory; Nevertheless, doubling the tempo did not prevent subjects from precisely targeting the note.

Depending on the correction reactiveness setting, it is interesting to investigate when the correction is actually applied. The number of notes effectively corrected is smaller than the number of played notes for both correction settings, as reported in Table II. Approximately half the notes were corrected in total. The fast correction had a critical time of 100 ms and a transition time of 50 ms, which equated to the correction being effective after 150 ms. The slow correction came into effect only after 250 ms. At the fast tempo (240 b.p.m.), a beat occurred every 250 ms. This leaves little time for the correction to be applied after stabilization of the note. Consequently, the settings define a tempo limit above which the correction is not likely to apply. Furthermore, some subjects did not play the notes in a stable enough manner to trigger correction.

4.2.4. Onset Correction and Contour Correction. The same experimental protocol was used in a previous study investigating the question of accuracy and precision for *staccato* notes, i.e., with a separate stylus-to-tablet contact for each note [Perrotin and d'Alessandro 2013]. In that study, only the Onset Correction was applied (no Contour Correction). The study also used a different set of melody (5-note) patterns. Two of the current subjects also took part in the previous study. The accuracy and precision results for that *staccato* experiment are reported in Figure 9. Each box represents the accuracy and precision for all subjects. The results labeled *legato stimuli* are from the current study and represent the output values with and without Contour Correction in effect (fast and slow corrections combined). The results labeled *staccato stimuli* are the results of the previous study and also represent the output values both with and without Onset Correction in effect. For accurate comparison with the previous experiment, only one value for accuracy and precision was computed for each subject and all trials were combined.

Although the difference in accuracy with no correction is not significant between the *staccato* and *legato* conditions, the *legato* precision with correction is significantly higher than the *staccato* precision without correction ($W = 65, p < 0.01$). This may be explained by the difficulty of the tasks. All the melodies presented in the *staccato* experiment had a similar shape: an upward interval, downward third, upward third, and downward interval, and they were symmetrical. The melodies of the current *legato* experiment were both longer and more complex. With correction, accuracy and precision are not significantly different. This result indicates that correction conferred the same improvement in both experiments and did not depend on task difficulty. Moreover, the correction procedure was shown to work equally well for both *staccato* and *legato* melodies.

Fig. 10. Subjects' MOS rate depending on three correction settings: no correction, fast correction, or slow correction.

In summary, the corrected pitch precision is close to or lower than pitch difference limens. However, the method has two limitations: minimum precision and stability are required for effective Contour Correction. The user has to target each note with an error lower than 0.5ST, which may require some minimum level of experience with the instrument. Moreover, pitch stability during the correction can be achieved only when the player has a musical intention (i.e., targeting stable notes, not simply targeting a point on the tablet using glissandos and jerky motions) and when the tempo is sufficiently slow to allow time for the correction to be applied.

4.3. Experiment 2: Perceptual analysis

The objective evaluation of pitch accuracy and precision showed accurate and precise results when the correction was applied. Objective measures of pitch accuracy are strongly correlated to human perception of pitch accuracy [Larrouy-Maestri et al. 2013]. However, it seems of interest to directly study the effect of DPW correction on pitch accuracy perception.

4.3.1. Protocol. A Mean Opinion Score (MOS) paradigm was chosen to evaluate the subjects' perceptual accuracy of intonation. An MOS provides a global appreciation of relatively complex and long stimuli. Other experimental paradigms could have been used to evaluate the accuracy of individual notes or intervals, but in the case of melodies, a global judgment is more relevant. The subjects' task consisted of listening to audio recordings of stimuli played under different conditions and rating their pitch accuracy on a MOS scale from 1 (poor) to 5 (excellent). Subjects were asked to consider only pitch accuracy while ignoring other aspects of the recordings. Although the voice synthesis could have introduced a bias because it is lower in quality than a real voice, all of the stimuli used the same synthesis (see accompanying sound files) with only small variations between them and thus should be comparable in terms of sound quality.

The stimuli for this experiment were the melodies recorded by all subjects but one during Experiment 1. One subject was discarded because the very poor musicality of his results would likely have interfered with the pitch accuracy perception. Only the first of the three trials recorded by each subject for each condition were kept. Therefore, 216 stimuli were available (24 conditions $\times$ 9 subjects). These stimuli were presented in random order over two Genelec monitoring loudspeakers in an acoustically insulated and treated room designed for perceptual experiments.

In total, 9 subjects (2 female, average age 33 years) participated in the experiment. All had musical experience (average 21 years). Three of had also participated in Experiment 1. To distinguish subjects between Experiments 1 and 2, they will be termed Players (Experiment 1) and Listeners (Experiment 2) respectively. All Listeners were given prior instruction on the task and a participated in a training session in which they listened to 20 of the selected stimuli that reflected the range of pitch accuracies encountered for each melody. The protocols for the training session and the subsequent experiment were identical. Each subject required approximately 30 min to complete the test.

4.3.2. Results. The rates of Listeners' MOS for all melodies were computed for each condition. For instance, without correction, 28% of the melodies had a MOS of 3. The Listeners' scores are shown in Figure 10 as a function of the correction type. Significant differences appeared between the "No correction" and "Fast correction" distributions

($W = 35686, p < 0.01$) and between the "No correction" and "Slow correction" distributions ($W = 35600.5, p < 0.01$). No significant differences were observed between the "Fast correction" and "Slow correction" distributions.

These measures show that Listeners tended to rate stimuli with corrections with higher scores compared to stimuli without corrections. This result demonstrates the significant perceptive improvement of pitch accuracy provided by DPW correction.

5. DISCUSSION AND CONCLUSIONS

Playing a continuous pitch DMI is made easier with the help of a pitch correction method. The pitch correction algorithms currently used for DMI and acoustic instruments fall into three categories: 1) magnet correction, where output pitch is obtained by attracting the input pitch towards the target, with a given convergence speed; 2) warping correction, where output is obtained by direct mapping of input pitch on the pitch target; and 3) dynamic warping correction, where the properties of the warping function depend on the dynamics of the input pitch contour.

5.1. Tradeoff between accuracy and expressivity

The Dynamic Pitch Warping correction method presented in this study was able to correct the pitch played on a continuous interface such as a pen tablet in real time. Experiments showed that the method could automatically adjust the pitch to perfectly tuned target notes with remarkable accuracy and precision. The differences relative to targets were less than human pitch difference limens. This was perceptually verified with a perceptive experiment. Regarding expressivity, the DPW method allows for pitch modulations that makes it possible to correct vibrato, portamento, or glissando without distortions. Depending on the style of music played, different parameter settings may be preferred.

For expressive modulations, the correction introduces small distortions in relatively slow performances. The dynamics of trajectory correction can be adjusted by a detection interval, a critical time, and a correction transition time. We suggest using a detection interval of I = 0.1 ST and that the critical time T_c should be adjusted depending on the tempo of the piece. Section 3.4 shows that T_c = 250 ms is a good candidate for small expressive distortions, but a higher T_c should be chosen for higher tempi. The lack of expressivity introduced by a higher critical time T_c is compensated for because musicians naturally play with less pitch expressivity in fast successions of notes.

Because no differences were found between the accuracy and precision of input notes for non-corrected and corrected stimuli, it appears that the correction does not foster laziness in performance, thus causing the player to target only the area around the note and relying on the correction. The process of learning the instrument does not appear to be impeded by the pitch correction algorithm.

The correction method has been implemented in the *Cantor Digitalis*. This software is available under a freeware license[9]. The correction algorithm, implemented in the Max programming language, can be retrieved from the *Cantor Digitalis* package[10]. This DMI has been successfully used in a number of public musical performances. In this context, the musicians appreciated the comfort brought by the DPW correction, particularly for music at the faster tempo. The robustness of the method has been demonstrated in the demanding situation of a concert. Equipped with the DPW algorithm, the *Cantor Digitalis* won the first price at the Guthman musical instrument

[9]https://cantordigitalis.limsi.fr/download_en.php last checked: October 19, 2015.
[10]The correction algorithm is located at:
CantorDigitalis1.1/Sources/LIMSI.CantorDigitalis/patchers/AccuracyCorrection_v4.0

competition 2015[11].

5.2. Target correction vs. Contour correction

In the visual domain, the *Sticky icons* approach seems to be preferable for target acquisition [Cockburn and Firth 2003], [Parker et al. 2005]. By changing the C-D ratio or gain between input and output variables, this method assumes that subtle cursor position modulations can be considered as noise. Only the end of the movement is considered relevant information, but not the manner in which that end position is reached.

In contrast, the *Expanding targets* method does not change the C-D ratio; instead, it changes the size of the target. Using an adaptive warping function, the DPW correction is somewhat similar to the *Expanding targets* correction, because the pitch target can be reached from any place in its vicinity. In a musical context, expanded targets do not overlap but are juxtaposed, similar to the piano-like keys represented on the visual references fixed to the tablet surface. With or without correction, the user sees only expanded targets, and no visual distraction is introduced when applying the correction. Therefore, the major drawback attributable to expanding targets in a visual context does not matter in a musical context using DPW correction. As such, although *Sticky icons* or similar methods are effective for target acquisition only, *Expanding targets* or DPW are more versatile because they are also effective for contour correction, which is an essential feature for musical applications.

5.3. Future directions

Though this study is based on an instrument that uses a graphic tablet with a pen and focuses on a musical context, it could be inspiring for different domains of applications and interfaces. The algorithm is well suited to situations where both accurate target acquisition and subtle motion analysis are required simultaneously. This will apply directly in devices similar to the pen tablet such as finger touch or free-hand pointing interfaces. Moreover, the DPW method is able to correct any continuous physical quantity that presents discrete and regular target values. For instance, in the context of vocal synthesis, a Dynamic Vowel Warping method could be implemented to target specific vowels among a continuous articulation space. In the visual domain, there are also situations where accuracy, precision, and expressivity are needed simultaneously in visual pointing tasks. For instance, in a drawing application, a correction algorithm could attract the line on a grid when it is being drawn at a slow rate but could also leave the artist free to make expressive contours when the stylus speed exceeds a given threshold. In hands-free applications, a target could be acquired together with the expressivity encapsulated in the target acquisition motion itself. Overall, it is easy to envisage uses for the DPW algorithm in these and other situations to improve user comfort.

A. LIST OF ACCOMPANYING AUDIO AND VIDEO FILES

[11] http://guthman.gatech.edu/2015winners last checked: October 19, 2015.

Table III. List of accompanying audio and video files

Name	Type	Correction	Setting	Playback speed	Reference
No Correction.wav	Audio	None		Normal	Figure 4
DPW_A_normal.mov	Video and audio	DPW	A	Normal	Table I and Figure 4
DPW_A_slow.mov	Video	DPW	A	Slow (1/3)	Table I and Figure 4
DPW_B_normal.mov	Video and audio	DPW	B	Normal	Table I and Figure 4
DPW_B_slow.mov	Video	DPW	B	Slow (1/3)	Table I and Figure 4
DPW_C_normal.mov	Video and audio	DPW	C	Normal	Table I and Figure 4
DPW_C_slow.mov	Video	DPW	C	Slow (1/3)	Table I and Figure 4
Haken_D.wav	Audio	Haken	D	Normal	Table I and Figure 4
Haken_E.wav	Audio	Haken	E	Normal	Table I and Figure 4
Haken_F.wav	Audio	Haken	F	Normal	Table I and Figure 4
Discrete.wav	Audio	Discrete		Normal	Figure 1 - second panel

REFERENCES

R. Balakrishnan. 2004. "Beating" Fitts' Law: Virtual Enhancements for Pointing Facilitation. *Int. J. Hum.-Comput. Stud.* 61, 6 (Dec. 2004), 857–874. DOI:http://dx.doi.org/10.1016/j.ijhcs.2004.09.002

R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. 2004. Semantic Pointing: Improving Target Acquisition with Control-display Ratio Adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 519–526. DOI:http://dx.doi.org/10.1145/985692.985758

O. Chapuis, J.-B. Labrune, and E. Pietriga. 2009. DynaSpot: Speed-dependent Area Cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1391–1400. DOI:http://dx.doi.org/10.1145/1518701.1518911

A. Cockburn and A. Firth. 2003. Improving the Acquisition of Small Targets. In *People and Computers XVII — Designing for Society*, Eamonn O'Neill, Philippe Palanque, and Peter Johnson (Eds.). Springer London, 181–196. DOI:http://dx.doi.org/10.1007/978-1-4471-3754-2_11

C. d'Alessandro, L. Feugère, S. Le Beux, O. Perrotin, and A. Rilliard. 2014. Drawing melodies: Evaluation of Chironomic Singing Synthesis. *The Journal of the Acoustical Society of America* 135, 6 (June 2014), 3601–3612. DOI:http://dx.doi.org/10.1121/1.4875718

S. Dalla Bella, J.-F. Giguère, and I. Peretz. 2007. Singing proficiency in the general population. *Acoustical Society of America* (November 22 2007), 1182–1189. http://www.brams.umontreal.ca/plab/downloads/Dalla-Bella_et_al._2007.pdf

L. Feugère. 2013. *Synthèse par règles de la voix chantée contrôlée par le geste et applications musicales*. Ph.D. Dissertation. Université Paris VI. http://hal.archives-ouvertes.fr/index.php?halsid=ru8c6b592ui2mm9dju61ip7tt3&view_this_doc=tel-00926980&version=1

V. Goudard, H. Genevois, and L. Feugère. 2014. On the playing of monodic pitch in digital music instruments. In *ICMC/SMC 2014*, Anastasia Georgaki and Giorgos Kouroupetroglou (Eds.). National and Kapodistrian University of Athens, Athènes, Greece, 1418–1425. https://hal.archives-ouvertes.fr/hal-01073240

A. Grell, J. Sundberg, S. Ternström, M. Ptok, and E. Altenmüller. 2009. Rapid pitch correction in choir singers. *The Journal of the Acoustical Society of America* 126, 1 (May 2009), 407–413. DOI:http://dx.doi.org/10.1121/1.3147508

L. Haken. 2009. Position correction for an electronic musical instrument. (Nov. 17 2009). http://www.google.com/patents/US7619156 US Patent 7,619,156.

L. Haken, E. Tellman, and P. Wolfe. 1998. An Indiscrete Music Keyboard. *Computer Music Journal* 22, 1 (1998), 30–48.

H.A. Hildebrand. 1999. Pitch detection and intonation correction apparatus and method. (Oct. 26 1999). http://www.google.com/patents/US5973252 US Patent 5,973,252.

R. Lamb and A. N. Robertson. 2011. Seaboard: a New Piano Keyboard-Related Interface Combining Discrete and Continuous Control. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*. 503–506. http://www.nime.org/proceedings/2011/nime2011_503.pdf

P. Larrouy-Maestri, Y. Lévêque, D. Schön, A. Giovanni, and D. Morsomme. 2013. The Evaluation of Singing Voice Accuracy: A Comparison Between Subjective and Objective Methods. *Journal of Voice* 27, 2 (2013), 259.e1 – 259.e5. DOI:http://dx.doi.org/10.1016/j.jvoice.2012.11.003

I. S. MacKenzie and S. Riddersma. 1994. Effects of output display and control—display gain on human performance in interactive systems. *Behaviour & Information Technology* 13, 5 (1994), 328–337. DOI:http://dx.doi.org/10.1080/01449299408914613

V. Mäkelä, T. Heimonen, and M. Turunen. 2014. Magnetic Cursor: Improving Target Selection in Freehand Pointing Interfaces. In *Proceedings of The International Symposium on Pervasive Displays (PerDis '14)*. ACM, New York, NY, USA, Article 112, 6 pages. DOI:http://dx.doi.org/10.1145/2611009.2611025

M. J. McGuffin and R. Balakrishnan. 2005. Fitts' Law and Expanding Targets: Experimental Studies and Designs for User Interfaces. *ACM Transactions on Computer-Human Interactions (TOCHI)* 12, 4 (Dec. 2005), 388–422. DOI:http://dx.doi.org/10.1145/1121112.1121115

A. P. McPherson, A. Gierakowski, and A. M. Stark. 2013. The Space Between the Notes: Adding Expressive Pitch Control to the Piano Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2195–2204. DOI:http://dx.doi.org/10.1145/2470654.2481302

B. C. J. Moore. 1973. Frequency difference limens for shortduration tones. *The Journal of the Acoustical Society of America* 54, 3 (1973), 610–619. DOI:http://dx.doi.org/10.1121/1.1913640

N. Orio, N. Schnell, and M. M. Wanderley. 2001. Input Devices for Musical Expression: Borrowing Tools from HCI. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*

(NIME '01). National University of Singapore, Singapore, Singapore, 1–4. http://dl.acm.org/citation.cfm?id=1085152.1085157

J. K. Parker, R. L. Mandryk, M. N. Nunes, and K. M. Inkpen. 2005. TractorBeam Selection Aids: Improving Target Acquisition for Pointing Input on Tabletop Displays. In *Human-Computer Interaction - INTERACT 2005*, MariaFrancesca Costabile and Fabi Paternò (Eds.). Lecture Notes in Computer Science, Vol. 3585. Springer Berlin Heidelberg, 80–93. DOI:http://dx.doi.org/10.1007/11555261_10

O. Perrotin and C. d'Alessandro. 2013. Adaptive mapping for improved pitch accuracy on touch user interfaces. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME) (NIME '13)*. 186–189. http://nime2013.kaist.ac.kr/program/papers/day2/paper5/178/178_Paper.pdf

P. Q. Pfordresher, S. Brown, K. M. Meier, M. Belyk, and M. Liotti. 2010. Imprecise singing is widespread. *The Journal of the Acoustical Society of America* 128, 4 (July 18 2010), 2182–2190. DOI:http://dx.doi.org/10.1121/1.347878

A. Pino, E. Tzemis, N. Ioannou, and G. Kouroupetroglou. 2013. Using Kinect for 2D and 3D Pointing Tasks: Performance Evaluation. In *Proceedings of the 15th International Conference on Human-Computer Interaction: Interaction Modalities and Techniques - Volume Part IV (HCI'13)*. Springer-Verlag, Berlin, Heidelberg, 358–367. DOI:http://dx.doi.org/10.1007/978-3-642-39330-3_38

O. Polacek, M. Klima, A. J. Sporka, P. Zak, M. Hradis, P. Zemcik, and V. Prochazka. 2012. A Comparative Study on Distant Free-hand Pointing. In *Proceedings of the 10th European Conference on Interactive Tv and Video (EuroiTV '12)*. ACM, New York, NY, USA, 139–142. DOI:http://dx.doi.org/10.1145/2325616.2325644

J. Sundberg. 1973. *Data on maximum speed of pitch changes*. Quarterly progress and status report. Royal Institute of Technologies - Dept. for Speech, Music and Hearing. 39–47 pages. http://www.speech.kth.se/prod/publications/files/qpsr/1973/1973_14_4_039-047.pdf

J. Sundberg. 1994. *Acoustic and Psychoacoustic Aspects of Vocal Vibrato*. Quarterly Progress and Status Report 2-3. Royal Institute of Technologies - Dept. for Speech, Music and Hearing. 45–68 pages. http://2.inarchive.com/1206/14/155/2iQPD9.pdf

A. Worden, N. Walker, K. Bharat, and S. Hudson. 1997. Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. ACM, New York, NY, USA, 266–271. DOI:http://dx.doi.org/10.1145/258549.258724

Y. Xu and X. Sun. 2000. How Fast Can We Really Change Pitch? Maximum Speed of Pitch Change Revisited. In *Proceedings of The International Conference on Spoken Language Processing*. 666–669. http://www.haskins.yale.edu/yixu/00571.pdf

Received February 2007; revised March 2009; accepted June 2009