# REFINING FUNDAMENTAL FREQUENCY ESTIMATES USING TIME WARPING

*Fabian-Robert Stöter, Nils Werner, Stefan Bayer, and Bernd Edler*

International Audio Laboratories Erlangen*
Am Wolfsmantel 33, 91058 Erlangen, Germany

## ABSTRACT

Algorithms for estimating the fundamental frequency ($F0$) of a signal vary in stability and accuracy. We propose a method which iteratively improves the estimates of such algorithms by applying in each step a time warp on the input signal based on the previously estimated fundamental frequency. This time warp is designed to lead to a nearly constant $F0$. A refinement is then calculated through inverse time warping of the result of an $F0$ estimation applied to the warped signal. The proposed refinement algorithm is not limited to specific estimators or optimized for specific input signal characteristics. The method is evaluated on synthetic audio signals as well as speech recordings and polyphonic music recordings. Results indicate a significant improvement on accuracy when using the proposed refinement in combination with several well-known $F0$ estimators.

***Index Terms***— Fundamental frequency estimation, pitch tracking, pitch estimation, time warping

## 1. INTRODUCTION

An estimate of the fundamental frequency $F0$ of a signal is required in various applications of audio and speech signal processing. $F0$ is often synonymously referred to as pitch which is a perceptual measure. In the past, a number of algorithms were presented to provide such estimates, with many of them being designed for specific applications. Some scenarios are targeted to extract the fundamental frequency of the predominant source [1] in a mixture of other sources. In other applications, algorithms are used to extract fundamental frequencies of multiple sources simultaneously present in a signal [2]. However, the most common scenario in many works is to extract the fundamental frequency of a monophonic and harmonic audio signal containing speech or music [3–9].

The development of novel methods for fundamental frequency estimation, performing as well as earlier methods, such as the popular correlation based YIN algorithm [5], has proven challenging. In a recent study [10] it is stated that YIN still clearly performs best in terms of accuracy. Nevertheless, when using YIN or other block based algorithms, a frame length and a hop size have to be selected trading temporal resolution on one side against frequency accuracy and robustness on the other side.

Especially when the signal is polyphonic, the robustness is the most crucial aspect of a pitch estimator. In recent work from Mauch et al. [11], the robustness of the YIN algorithm is improved by probabilistic post-processing. However, besides robustness, there is a variety of use cases requiring high accuracy as well as high temporal resolution. Application in parametric audio coding [12] requires the parameterization of pitch bends and vibratos. Furthermore, source separation algorithms aiming at the extraction of harmonic sources from the mixture can make use of an instantaneous $F0$ estimate [13, 14]. There are already contributions addressing the improvement of accuracy of $F0$ estimates such as [15] which introduced a non-integer similarity model or [9] which belongs to the group of parametric pitch estimators.

We propose to improve the output of already existing algorithms in terms of temporal resolution as well as accuracy by iterative time warping. Two other contributions already make use of time warping in the context of pitch estimation. Resch et al. [6] proposed an instantaneous pitch estimation technique which optimizes a warping function that would lead to a constant pitch signal. Their optimization framework minimizes a cost function specifically targeted for speech signals. Azarov et al. have introduced an improved version of RAPT (called iRAPT1 and iRAPT2) which also uses time warping to some extent [16] but misses an additional step as will be shown in Section 2.2.2. Our main contribution is a time warping based refinement method that is applicable to any F0 estimate. Our method emphasizes the strengths of different estimators and thus can even help to improve their robustness. In the following, we will describe the refinement method (Section 2) and show the experimental evaluation and its results (Section 3).

## 2. REFINED $F0$ ESTIMATION

Depending on the algorithm and application, there are several reasons why $F0$ estimators deliver a less than ideal performance. When the signal tested is not tonal — like in unvoiced parts of speech — a proper estimation is impossible. If the estimator is optimized on purely harmonic signals, inhar-

[*] A joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institut für Integrierte Schaltungen (IIS).

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

monicity or frequency jitter of the input signal will increase the estimation error. Many of these reasons will lead to errors on the coarse level of the estimate (like octave jumps). The fine level accuracy is mostly influenced by parameters like time and/or frequency resolution of the estimator. A signal containing rapid changes of the frequency or modulations like "vibrato" is therefore more affected regarding fine level error. To obtain a more accurate estimate, we propose to time warp the signal by using the coarse level estimate towards a more constant pitch. The underlying assumption here is that pitch estimators generally perform better the more constant the pitch is. In this section, we formulate the mathematical background of the time warping and present our proposed method for obtaining a refined $F0$ estimate.

### 2.1. Initial $F0$ estimate

The first step is to calculate an initial $F0$ estimate by using an existing pitch estimator. Note that we later require the estimate to be defined for every input sample, thus $F[n]$ may require interpolation. In our pipeline, we use linear interpolation for all estimators. $F0$ estimators, like YIN [5], also provide a measure of confidence $c[n]$.

### 2.2. Time warping and refinement

In this step, we apply *time warping* which refers to a strictly monotonous mapping of the natural or linear time scale $t$ to a warped time scale $\tau$ via a mapping function $\tau = w(t)$. The mapping between the two domains for the continuous time case then is:

$$\breve{x}(\tau) = x(w^{-1}(\tau)), \quad x(t) = \breve{x}(w(t)) \qquad (1)$$

where $x(t)$ is the linear-time signal and $\breve{x}(\tau)$ is the warped-time signal. For the discrete time case, the signals in both linear-time and warped-time domains are sampled using a constant sample interval $T$. With sample indices $\nu$ and $n$ for the warped-time domain and linear time-domain respectively, the warping is performed by

$$\breve{x}[\nu] = x(\sigma[\nu]) \qquad \text{with } \sigma[\nu] = w^{-1}(\nu T), \qquad (2)$$

and the inverse warping by

$$x[n] = \breve{x}(s[n]) \qquad \text{with } s[n] = w(nT). \qquad (3)$$

#### 2.2.1. *Warp contour*

In our application, the warp map $w(t)$ is constructed in such a way that the instantaneous changes in frequency of the signal in the linear time domain are minimized in the warped time domain. For this, we derive the map from an estimate of the fundamental frequency $F0$.

For processing, the actual information needed is not the absolute instantaneous fundamental frequency but only its change over time. This means that the warping contour can be derived from an algorithm which may differ from the actual $F0$ estimator.

The discrete time warp map $w[n]$ is the scaled sum of the relative frequency contour (the *warp contour*) $W[n]$:

$$w[n] = N \frac{\sum_{l=0}^{n} W[l]}{\sum_{k=0}^{N-1} W[k]} \qquad 0 \leq n < N, \qquad (4)$$

where $N$ being the number of samples of the signal under consideration. As stated above the full warp map $w(t)$ is then obtained by linearly interpolating $w[n]$. From the requirements for the mapping function it follows that $W[n]$ has to be greater than zero for all $n$. In the case of a perfect $F0$ estimate, the signal warped with the resulting contour would have a constant $F0$ equal to the average $\bar{W}$.

In the scope of this work, the warping is applied globally over the full length of the signals under consideration. An optional confidence measure $c[n]$ can be incorporated for a processed version of the warping contour. This ensures that the warp contour has no discontinuities that result in additional artifacts after re-sampling. If the estimator does not provide such a measure, a separate voiced/unvoiced detection algorithm can be used. To obtain a warp contour $W[n]$ from an $F0$ estimate we propose the following steps: **(A)** initialize the warp contour with $F0$ estimate $W = F$, **(B)** find contour segments with high confidence, i.e. $c[n]$ exceeds a given threshold, **(C)** linearly connect the high confidence contour segments and **(D)** set start and end of warp contour to a constant value if confidence is below threshold. That way warping according to $F0$ is applied in the regions of high confidence without significantly affecting the gaps in-between.

#### 2.2.2. *Obtaining a refined estimate*

To improve the accuracy of the $F0$ estimate, time warping is applied to the input signal $x[n]$ based on $W$. The input signal is 128-times oversampled using sinc based interpolation filters. From $\breve{x}[n]$ a new $F0$ estimate $\breve{F}_1[\nu]$ is being calculated as in step **(A)**. The first step therefore is similar to [6]. Additionally, a warped confidence measure $\breve{c}_1[\nu]$ can be used to convert $\breve{F}_1[\nu]$ into a warped *warp contour* $\breve{W}_1[\nu]$. It is possible to linearly add $\breve{F}_1[\nu]$ to the first estimate for refinement, as it is done in [16]. However for linear sweeps, the warped estimate is shifted in time. Thus an error is introduced which is even more distinct if the first $F0$ estimate is error prone. We therefore propose a method to reduce this error:

- Inverse time warping is applied to $\breve{F}_1[\nu]$ based on the original warp contour $W$ resulting in $F_1[n]$.
- A refined $F0$ estimate after one iteration is then calculated by $F_1^T[n] = F_1[n] \cdot W[n]/\bar{W}$ assuming that the warp contour is initialized as in step **(A)** above.

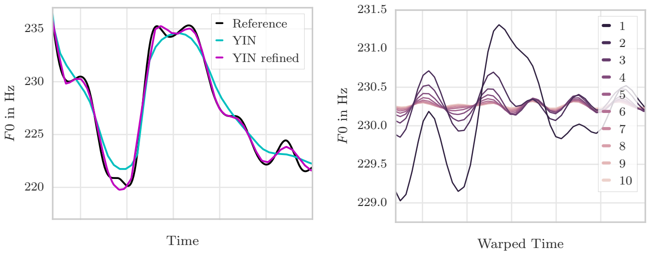

**Fig. 1**. $F0$ refinement for one excerpt of synthesized speech using **YIN** [5] with 10 iterations. *Left*: Estimated $F0$ in linear time. *Right*: estimates after each warping iteration in warped time.

- The refinement can be repeated $k$ times to obtain a better estimate. To avoid accumulating errors introduced by the re-sampling based warping, more iterations benefit from calculating a refined warp contour/warp map instead of doing a nested warping on the input signal. The map is obtained by inverse time warping of the warp contour $\breve{W}_1[\nu]$ resulting in $W_1[n]$. A refined warp contour $W_1^r[n]$ is then obtained in the same way as the refined $F0$ estimate is calculated. For the calculation of the $k$th step, time warping is based on the $W_{k-1}^r[n]$ refined warp contour.

An example of the proposed refinement is depicted in Figure 1. The final refined estimate is closer to the reference than the $F0$ estimator without refinement. It also shows (right plot) how much "flatter" the $F0$ contour becomes after each iteration. Note that compared to [6], our method does not use a complex optimisation scheme but relies on the performance of the pitch estimator in successive iterations. Hence our "black box" like post processing simplifies the procedure such that it can be applied to any pitch estimator. That way the selection of a pitch estimator which best fits to the signal type can be seen as an optimisation.

## 3. EXPERIMENTS AND EVALUATION

### 3.1. Estimators

For the evaluation of the proposed $F0$ refinement, we test the refinement algorithm with the following $F0$ estimators:
**YIN** [5] is used as an FFT based implementation [17]. The confidence measure is thresholded for values lower than 0.6 on the speech recordings. **iRAPT1,2** [16] are improved versions of the RAPT framework. We use the author's MATLAB implementation of the iRAPT1 and iRAPT2 algorithms. iRAPT2 is a refinement method that is comparable to our proposed method. To evaluate the results, we apply our refinement to iRAPT1 and compare it with the refinement produced by iRAPT2. $c[n] < 0.7$ is used for thresholding speech recordings. **MELODIA** [1] is not designed to be an $F0$ estimator but is able to extract the *predominant* melody in a polyphonic mixture. We increase the bin resolution to 0.5 semitones, to increase the accuracy. We used the ESSENTIA implementation. For thresholding we use the built-in voiced/unvoiced detection. For YIN and MELODIA, we evaluate on a frame length of 64 ms and a hop size of 16 ms. For iRAPT1 and iRAPT2 we use the fixed frame length parameters of the author's implementation.

### 3.2. Evaluation

We use the established evaluation measures GROSS PITCH ERROR (GPE) and MEAN FINE PITCH ERROR (MFPE) [16]. We focus on MFPE in our results, measuring the absolute deviation of $F0_{\mathrm{true}}$ and the $F0$ estimate per sample. As mentioned in [6], evaluating the accuracy of $F0$ estimates is challenging because of the lack of ground truth datasets annotated on a time scale with such a high resolution. Most of the available audio test data sets are not suitable because the $F0$ annotation is only available with low time resolutions. By using such a dataset there is a risk that the refined $F0$ estimate is higher in MFPE. This is because the refined estimates show more of the fine structure deviating from the coarse annotation which then is considered as piecewise constant. To address this issue, we first present the evaluation results on synthetic data. To verify our synthetic results, we present the results of speech data annotated on 10 ms frames derived from laryngograph signals. We did only evaluate and process the voiced parts of the signals as indicated in the provided annotation labels. Also note that since we focus on the MFPE, all segments where one of the estimators results in a GPE $> 0$ are excluded from the results, hence the GPE for all of our results is 0. The proposed refinement has been processed with one iteration ($k = 1$). Experiments showed that more iterations only marginally improve the results.

#### 3.2.1. *Oracle Refinement*

Since the proposed refinement algorithm repeatedly applies pitch estimation, the performance of these estimators on the time warped (nearly constant) signal is of interest. Therefore we included the results of an oracle refinement where the first estimate is set to a ground truth pitch. Additionally this also does reveal information about the quality of the ground truth annotation itself.

#### 3.2.2. *Synthetic Data*

To generate synthetic test data we use pitch label annotations of the PTDB-TUG speech data set [18]. We synthesize the melody or voice using a simple sinusoidal signal model. To get accurate ground truth data, the pitch annotations were upsampled to audio rate by using linear interpolation for the PTDB-TUG. Similar to [11], we then synthesized the data using cosine based oscillators adding 10 harmonics to each signal output. The test set has been rendered at 16 kHz. The complete PTDB-TUG set results in almost 10 hours of input signal data. We present the results of the synthetic data

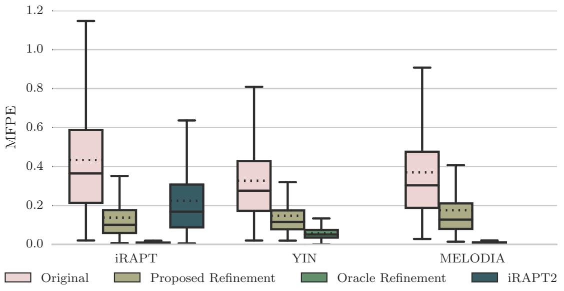

**Fig. 2**. Results from the synthesized PTDB-TUG dataset. MFPE grouped by estimator. Solid/dotted lines represent medians/means. Outliers are not shown.

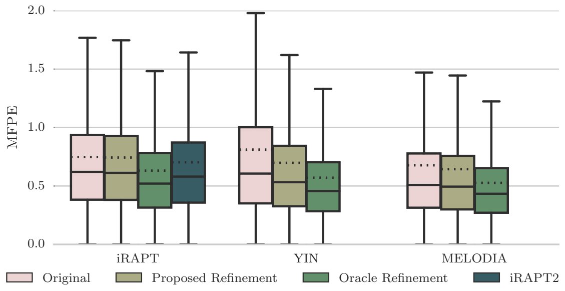

**Fig. 3**. Results from the real recordings PTDB-TUG dataset. MFPE grouped by estimator. Solid/dotted lines represent medians/means. Outliers are not shown.

as box plots in Figure 2 grouped by estimator. It shows that all estimators benefit from the refinement in terms of MFPE. The iRAPT1 estimator shows the best improvement of 68% in MFPE. As expected, the Oracle Refinement is almost at 0 MFPE.

### 3.2.3. *Speech Data*

For the results of the algorithm on real data we first used the same PTDB-TUG items as in the synthetic data but processed the accompanying speech recordings. The MFPE values were then calculated by averaging the sample wise $F0$ estimates from our proposed method over frame lengths of 10 ms to match the annotation data. The results are shown in Figure 3. The mean values indicate that the MELODIA algorithm performs best overall. We can see that the refinement does not show a clear effect on the iRAPT estimator. The oracle refinement results indicate that even if a ground truth is known, the refinement based on the warped (constant) signal can not get much lower in MFPE. As also seen on synthetic data, iRAPT2 does not show any significant improvements compared to our proposed refinements.

### 3.2.4. *Polyphonic Mixtures*

Pitch estimation of polyphonic mixture input signals in general is known to be more difficult than on monophonic signals. To show that our proposed refinement is not bound to the optimisation on specific signals we processed the MedleyDB [19] which consists of 108 professionally recorded music mixes where the main melody has been annotated by humans. We only evaluate the MELODIA [20] estimator in this scenario. Frame lengths and hop sizes were increased to 92 ms and 23 ms, respectively. The set is processed at 44.1 kHz. To further back up the results of the fine pitch error in this scenario, we additionally evaluated the results of a correlation based measure as introduced in [6] (See Equation (19)). Instead of computing the correlation coefficients on the mixture, we used the accompanying multi-tracks. The track which most predominantly contributed to the main melody has been chosen for the correlation coefficient measure. The results of the experiment are shown in Figure 4.

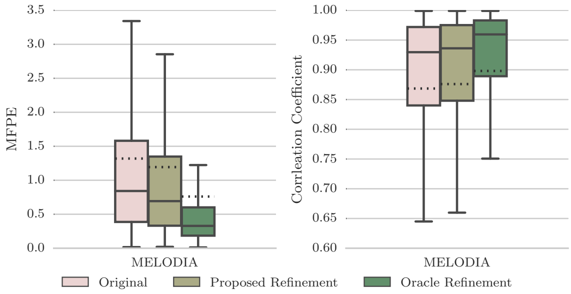

**Fig. 4**. Results from the real recordings MedleyDB dataset. MFPE and Correlation Coefficient grouped by estimator. Solid/dotted lines represent medians/means. Outliers are not shown.

### 3.2.5. *Statistical Analysis*

To test whether the refinements show statistical significance, paired Wilcoxon signed-rank tests for each of the estimators have been calculated. The test results in Table 1 show if there is a statistically significant ($\alpha = 0.05$) difference between the refined and unrefined groups on the voiced segments. For synthetic data as well as for speech, YIN and MELODIA show statistically significant improvements. However, the effect size $r$ on speech is lower than for synthetic data. The results for the MedleyDB are significant both in MFPE and correlation coefficient.

## 4. CONCLUSION

In this work we presented a method to improve the accuracy of $F0$ estimators. The proposed method uses time warping iteratively based on an initial $F0$ estimate. Therefore the implementation can be applied to any $F0$ estimator. We showed that by inverse time warping of a derived warp contour, an improved estimate is constructed. The algorithm is evaluated

| Estimator | Measure | $T$ | $p$ | $r$ |
|---|---|---|---|---|
| **PTDB-TUG (synthetic)** $n = 11487$ | | | | |
| YIN | MFPE | 55466 | .000 | 0.998 |
| iRAPT1 | MFPE | 13831 | .000 | 1.000 |
| MELODIA | MFPE | 12349 | .000 | 1.000 |
| **PTDB-TUG (speech)** $n = 9271$ | | | | |
| YIN | MFPE | 13082927 | .000 | 0.391 |
| iRAPT1 | MFPE | 21144487 | .186 | 0.016 |
| MELODIA | MFPE | 20484251 | .000 | 0.047 |
| **MedleyDB** $n = 1055$ | | | | |
| MELODIA | MFPE | 134380 | .000 | 0.517 |
| MELODIA | Correlation | 124390 | .000 | 0.553 |

**Table 1**. Result of Wilcoxon signed-rank test $T$, $p$ and effect sizes $r$.

on synthesized as well as real recordings. We compared our proposed refinement with an improved version of the RAPT framework and showed that the proposed refinements result in an improvement of up to 68% on synthetic data and 14% on speech in terms of the MFPE measure. The proposed method does also work on polyphonic signals without further optimisations. In future work, combinations of different $F0$ estimators in each step could be used to better balance the tradeoff between robustness and accuracy per application and scenario.

## REFERENCES

[1] J. Salamon and E. Gómez, "Melody Extraction From Polyphonic Music Signals Using Pitch Contour Characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, Aug 2012.

[2] A. P. Klapuri, "Multiple Fundamental Frequency Estimation Based on Harmonicity and Spectral Smoothness," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003.

[3] D. Talkin, "A Robust Algorithm for Pitch Tracking (RAPT)," in *Speech Coding & Synthesis*, W.B. Klejn and K.K. Paliwal, Eds., pp. 495–518. Elsevier, 1995.

[4] P. Boersma, "Praat, a System for Doing Phonetics by Computer," *Glot international*, vol. 5, no. 9/10, pp. 341–345, 2001.

[5] A. De Cheveigné and H. Kawahara, "YIN, a Fundamental Frequency Estimator for Speech and Music," *The Journ. of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[6] B. Resch, M. Nilsson, A. Ekman, and B. W. Kleijn, "Estimation of the instantaneous pitch of speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 813–822, 2007.

[7] A. Camacho, *SWIPE: A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music*, Ph.D. thesis, University of Florida, 2007.

[8] D. Tidhar, M. Mauch, and S. Dixon, "High precision frequency estimation for harpsichord tuning classification," in *2010 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 61–64.

[9] M. G. Christensen, A. Jakobsson, and S. H. Jensen, "Joint high-resolution fundamental frequency and order estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1635–1644, 2007.

[10] O. Babacan, T. Drugman, N. D'alessandro, N. Henrich, and T. Dutoit, "A Comparative Study of Pitch Extraction Algorithms on a Large Variety of Singing Sounds," in *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 7815–7819.

[11] M. Mauch and S. Dixon, "pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 659–663.

[12] H. Purnhagen and N. Meine, "HILN -The MPEG-4 Parametric Audio Coding Tools," in *IEEE Int. Symposium on Circuits and Systems 2000*. IEEE, 2000, vol. 3, pp. 201–204.

[13] T. Virtanen, A. Mesaros, and M. Ryynänen, "Combining Pitch-Based Inference and Non-Negative Spectrogram Factorization in Separating Vocals From Polyphonic Music," in *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audition (SAPA2008)*, 2008, pp. 17–22.

[14] F. Stöter, S. Bayer, and B. Edler, "Unison Source Separation," in *17th Int. Conference on Digital Audio Effects (DAFx)*, 2014, pp. 235–241.

[15] Y. Medan, E. Yair, and D. Chazan, "Super Resolution Pitch Determination of Speech Signals," *IEEE Transactions on Signal Processing*, vol. 39, no. 1, pp. 40–48, 1991.

[16] E. Azarov, M. Vashkevich, and A. Petrovsky, "Instantaneous Pitch Estimation Based on RAPT Framework," in *20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 2787–2791.

[17] D. et. al. Bogdanov, "Essentia: An Audio Analysis Library for Music Information Retrieval," in *14th Int. Conf. on Music Information Retrieval (ISMIR)*, 2013, pp. 493–498.

[18] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "A Pitch Tracking Corpus with Evaluation on Multipitch Tracking Scenario," in *Interspeech 2011*, pp. 1509–1512.

[19] R. M. Bittner et al., "Medleydb: A multitrack dataset for annotation-intensive MIR research," in *15th Int. Society for Music Information Retrieval Conference ISMIR, Taipei, Taiwan, October 27-31, 2014*, Hsin-Min Wang, Yi-Hsuan Yang, and Jin Ha Lee, Eds., 2014, pp. 155–160.

[20] J. Salamon, E. Gomez, D. P. Ellis, and G. Richard, "Melody extraction from polyphonic music signals: Approaches, applications, and challenges," *IEEE Signal Processing Mag.*, vol. 31, no. 2, pp. 118–134, 2014.