# A Music Stream Segregation System Based on Adaptive Multi-Agents

Kunio Kashino Hiroshi Murase
NTT Basic Research Laboratories
3-1 Morinosato-Wakamiya, Atsugi-shi,
Kanagawa, 243-01, JAPAN
kunio@ca-sunl.brL.ntt .co.jp, murase@apollo3. brL.ntt.co.jp

## Abstract

A principal problem of auditory scene analysis is stream segregation: decomposing an input acoustic signal into signals of individual sound sources included in the input. While existing signal processing algorithms cannot properly solve this inverse problem, a multi-agent-based architecture has been considered to be a promising methodology in its modularity and scalability. However, most attempts made so far depend on subjectively defined rules to deal with variability of sounds. Here we propose a quantitatively principled architecture in agent interaction by formulating the problem as least-squares optimization. In this architecture, adaptation of the agents is the essential idea. We have developed two kinds of processing to realize adaptivity: template filtering and phase tracking. These mechanisms enable each agent to optimally, in the least-squares sense, track the individual sound. As an example application of the proposed architecture, we have built a music recognition system that recognizes instrument names and pitches of the notes included in ensemble music performances. Experimental results show that these adaptive mechanisms significantly improve the recognition accuracy.

## 1 Introduction

In recent years scene analysis based on acoustic information, termed auditory scene analysis, has received a renewal of interest. Recognizing external events based on acoustic information is an essential function for systems that work in the real world.

A principal problem toward auditory scene analysis is stream segregation [Bregman, 1990]. This segregation means decomposing an input signal into signals of individual sound sources included in the input. However, once multiple acoustic signals are mixed up, their segregation is, so far, considered very difficult because it is an ill-posed inverse problem.

Nevertheless, technical and applicational importance has attracted researchers to this field of study. Specifically, works intended to model integration of bottom-up and top-down processing includes [Lesser *et* a/., 1993], [Nakatani, *et* a/., 1995], and [Ellis, 1996]. These works are characterized by their architectures based on processing modules with simplified functions and communications between these modules, which we call a multi-agent architecture. While the architecture intrinsically enjoys modularity and scalability, quantitative background for behavior of agents is not yet established. Practically, the multi-agent based systems mentioned above require subjectively defined rules to control interaction schemes or to adjust parameters for modules in order to deal with variations of sounds.

Here we propose a quantitatively principled architecture, called Ipanema, designed to solve the stream segregation problem for sound mixtures. The essential idea is adaptation of agents to cope with variation of a sound. We have developed two kinds of processing in order to realize adaptivity: template filtering and phase tracking. These mechanisms enable each agent to optimally, in the least-squares sense, track the individual sound. As an example application of the proposed architecture, a music recognition system has been built. The evaluation tests show that the adaptive mechanisms have significantly improve the recognition accuracy in comparison to a conventional signal-detection/separation method based on the matched filtering.

In the following part of this paper, Section 2 focuses the discussion on the adaptive processing, which is an essential part of the architecture. Section 3 then describes general configuration of the Ipanema architecture. After Section 4 introduces evaluation results of the recognition accuracy, Section 5 discusses implications of the present work in the context of existing related approaches. Finally Section 6 concludes the paper with the expected future work.

## 2 Adaptation of Templates

### 2.1 Template Filtering

We consider representing an input acoustic signal $z\{k)$ with a sum of template waveforms $y_n(k),$ where n is


Figure 1: A sound source model that consists of a template $r$ and an FIR filter $H$. Here, $H$ modifies the waveform of the original template $r$ to cope with variation of a sound.

the index that corresponds to each sound source and $k$ enumerates sampling time sequences. Our problem can be formulated as minimization of $J$ in the equation:

$$J = E\left[\left\{z(k) - \sum_{n=0}^{N-1} y_n(k)\right\}^2\right] , \qquad (1)$$

where $N$ is the estimated number of sound sources, which is not predefined, and $E$ denotes average over time. For $y_n(k)$, we employ one of the simplest models as depicted in Figure 1. Mathematically, the model can be written as

$$y_n(k) = \sum_{m=0}^{M-1} h_n(m)\, r_n(k-m) , \qquad (2)$$

where $h$ is an impulse response of the filter $H$, $r$ is a template waveform, and $M$ is the length of the impulse response, that is, the number of taps when one use the FIR filter as $H$.

In this formulation, one cannot predetermine the fixed sets of $h$ and $r$, because there is a diversity of waveforms even for one specific sound source. For the example of musical instruments, both top two waveforms, (a) and (b), in Figure 2 are piano sounds. Even if we ignore the phase information and consider only spectral power representations, the situation is essentially the same because there are a variety of spectrum patterns for one sound source. Therefore we need an adaptive mechanism. Here we would change $h_n(m)$. Equation (1) is rewritten using Equation (2) as

$$J = E\left[\left\{z(k) - \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} h_n(m)\, r_n(k-m)\right\}^2\right] . \qquad (3)$$

The necessary condition for $J$ to hold the minimum value over $h_n(m)$ is that the values of partial derivative $\partial J/\partial h_n(m)$ are 0 for all $n$ and $m$. Using this condition, it is straightforward to derive $N \times M$ pieces of simultaneous linear equations as follows,

$$\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} E\left[r_i(k-j)\, r_n(k-m)\right]\, h_n(m) = E\left[r_i(k-m)\, z(k)\right] , \qquad (4)$$

where $i = \{0, 1, \cdots, N-1\}$ and $j = \{0, 1, \cdots, M-1\}$. Since the number of equations $(N \times M)$ equals the number of unknown parameters ($h_n(m)$), the problem is reduced to calculation of the inverse matrix, which is a simple algebraic operation.

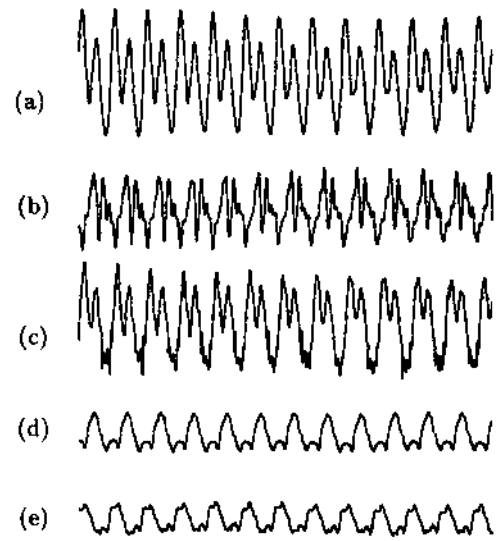


Figure 2: Template filtering. (a): an input signal, which is an F4 note of Yamaha's piano, from 160ms through 195ms from its onset. (b): the original piano template, which is Boesendolfer's, with the same pitch and the same time portion as (a). Though both (a) and (b) are piano sounds, their waveforms are different. (c), (d), and (e): piano(=(b)), flute, and violin templates processed by the template filtering (Number of filter taps = 160, sampling frequency = 48 kHz). The filtering modifies the (b) to yield the waveform (c), which has higher correlation with (a), than correlation between (d) and (a), and correlation between (e) and (a).

## 2.2 Phase Tracking

The condition for the above optimization scheme to be effective is that the fundamental frequency of each template r is exactly the same as the one included in $z$. This is because a linear filter, $H$, cannot change the frequency of an input signal. Therefore we need a phase tracking (i.e. instantaneous frequency tracking) method, which changes the phase of template r in accordance with the phase of the corresponding sound source signal included in the input signal $z$.

If the input signal is not a mixture of multiple sounds but a single sound, adaptive pitch tracking methods already invented can be used [Nehorai and Porat, 1986]. However, such signal processing methods are not directly applicable to a sound mixture where multiple pitches are present. Thus we have devised a simple algorithm to realize the phase adaptation. The algorithm consists of the following six steps.

(1) Perform frequency analysis to the input z, to extract fundamental-frequency components. Because $z$ may be a mixture of multiple sound signals, there may

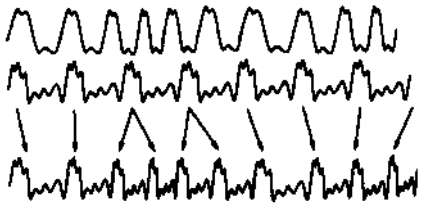

Figure 3: Waveforms demonstrating function of adaptive phase tracking. Top: input waveform $z$; middle: template without adaptive phase tracking; and bottom: template with adaptive phase tracking. The waveform in the bottom panel is used as $r_i(k)$ in Equation (4).

be multiple fundamental-frequency components.

(2) For each fundamental frequency component, choose $r_i$ that is a possible template of a sound included in $z$.

(3) Apply a narrow-band bandpass filter to $r_i$, using average fundamental frequency of each $r_i$ as the center frequency of the bandpass filter. For each time sample, store the phase of the output waveform of the bandpass filter. Let $p_{r,i}(k)$ denote the phase at time $k$.

(4) Apply the same bandpass filter, as applied to $r_i$, to the input $z$, and store the phase information for each fundamental frequency as $p_{z,i}(k)$.

(5) Calculate the required time shift $\Delta k_{r,i}(k)$. Because the phase difference $\Delta p_{r,i}(k)$ is given as

$$\Delta p_{r,i}(k) = p_{z,i}(k) - p_{r,i}(k), \qquad (5)$$

the time shift $\Delta k_{r,i}(k)$ is calculated by

$$\Delta k_{r,i}(k) = \frac{f_s}{2\pi f_{c,i}} \Delta p_{r,i}(k), \qquad (6)$$

where $f_s$ is the sampling frequency and $f_{c,i}$ is the center frequency of the applied bandpass filter.

(6) The amplitude value $r_i$ at time $k$ is given as

$$r_i(k) = r_i(\, k - \Delta k_{r,i}(k)\,). \qquad (7)$$

Figure 3 shows how this algorithm works.

## 3 Ipanema Architecture

This section introduces the configuration of the system architecture designed for calculating the scheme described in the previous section.

### 3.1 Overview

A specific feature of the stream segregation problem here is that "signals" or "noises" are not uniquely defined in advance; that is, a stream segregation system is required to handle multiple kinds of "signals" simultaneously[1].

[1]This is a contrast to speech recognition systems where it is defined that the "signal" is a human speech and "noises" are the sounds other than the speech.

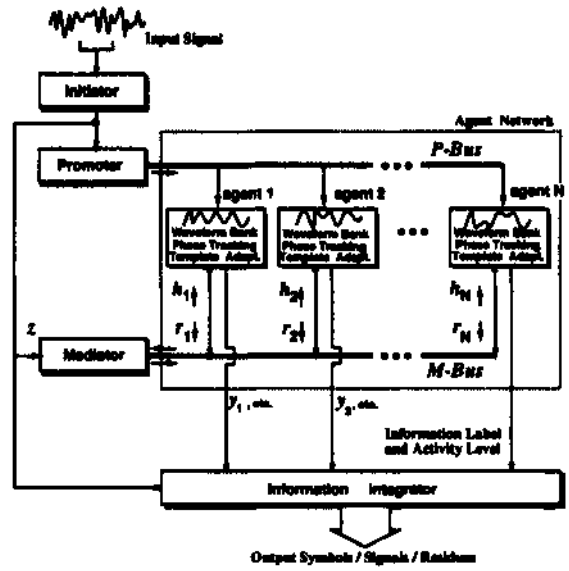


Figure 4: Configuration of Ipanema architecture.

Therefore it is a natural idea to build a stream segregation system by accumulating processing modules, each of which tries to extract a specific target sound as a "signal" in its charge, regarding other sounds as "noises". In addition, it is obvious that the modules should affect each other in order to create a valid interpretation of scenes.

Thus we propose a system architecture based on the multi-agent scheme, as shown in Figure 4. As an input, the system is fed with a sound signal that is a mixture of signals from multiple sound sources. The current version of the system assumes that the input is an ensemble music signal. As outputs, the system creates a symbolic representation that is similar to the musical scores, and waveforms produced by agents.

The system consists of the following elements: initiator, promoter, agent network, mediator of agents; therefore we call the architecture *Ipanema*. There is also a post-processing module called an information integrator.

### 3.2 Processing Modules

#### Initiator

The initiator cuts an input signal into frames and sends the waveform of each frame to succeeding processing. The frame length is variable; when an input signal is available, the initiator tries to find an onset of the sound. Every time it finds the onsets, the initiator creates a new frame.

#### Promoter

The promoter performs frequency analysis on the waveform sent from the initiator and extracts possibly-multiple fundamental-frequency- components included

in the frame. Then it outputs the fundamental frequencies to P-Bus (promotion bus), which is observed by all the agents.

### Agents

In our architecture, each agent in the "agent network" is a processing module that corresponds to a single sound source (flute, for example). Each agent has a bank of raw-template waveforms, each of which corresponds to a specific pitch and expression.

Each agent examines the P-Bus information to check the possibility that the sound source corresponding to the agent is included in the input or not. If the agent infers that there is a possibility of being included, the agent suggests waveform $r_i$, applying the adaptive phase tracking method described in Section 2 to one of the raw-template waveforms. If the agent finds little possibility of presence of the corresponding sound, it just keeps silence.

Waveforms $r_t$ are written to the common place called M-Bus (mediation bus) and passed to the mediator. Then the agents wait for the mediator to feed back answers. The answers from the mediator are sets of filter coefficients that optimally modify $r_i$ Each agent reads the answer from the mediator via M-Bus and then calculates an FIR filter with the answered coefficients to obtain a waveform $y_i$. The final output of the agent is waveform $y_i$, its average power $E\{y^2_i\}$ as activity level, and an information label that the agent is given; for example, "Flute C4".

In the current implementation, the agents in the agent network only communicates with the mediator. However, we expect that the extension of processing scheme would enable communication among the agents themselves in the agent network through M-Bus.

### Mediator

Here mediation of agents is formulated and reduced to the problem of matrix calculation, as discussed in Section 2. Thus the mediator first receives the r, via M-Bus from agents. Then it calculates the optimal filter coefficients for each agent, using Equation (4). Finally the mediator sends the coefficients back to each agent using M-Bus.

### Information Integrator

The information integrator is a post processing module that revises the symbolic output of the system. It receives an information label (e.g. Flute C4) and activity level from each agent, and basically, the label with the highest activity level for each note composes a symbolic version of output of the system. It is inevitable, however, that noises occasionally appear without higher-level information such as temporal or simultaneous relations between sounds. Thus the Bayesian networks are employed here in order to integrate multiple sources of information.

In the current system, note transition information has been introduced. The information integrator first constructs the Bayesian networks where nodes encode probabilities for the information labels and links represent temporal relation between the nodes. The integrator then updates the probabilities for the labels based on the probability propagation scheme [Pearl, 1986], integrating note transition statistics stored in advance. This paper, however, focuses on the adaptive mechanisms and the details on the information integrator will be reported in a separate article [Kashino and Murase, 1997].

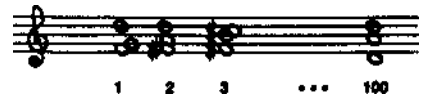


Figure 5: Test patterns used in note-recognition benchmark tests. Note that each chord includes a perfect fifth interval (i.e. 2:3 fundamental frequencies), making the recognition more difficult than a completely-random interval pattern.

## 4 Evaluations

We performed two kinds of tests to evaluate the system: a benchmark test for musical note recognition and a sample song recognition test. In both tests, the information integrator was *turned off* (i.e. the note transition information was not integrated) in order to evaluate basic performances of the adaptive mechanisms.

### 4.1 Benchmark Test

To evaluate the advantages of adaptive processing described in Section 2, we tried to conduct the benchmark test of note recognition used in [Kashino, *et a*/., 1995a; 1995b].

The test signal was a three-simultaneous-notes pattern, as shown in Figure 5. The pattern was composed and created by a computer using digitized acoustic signals (16bit, 48kHz) of natural musical instruments (flute, piano, and violin). We first recorded the single notes of those instruments at a recording studio and stored the waveforms on a computer. We then mixed the stored waveforms, selecting a designated number of notes. The selection of the notes were programmed to produce the *Class-2* note pattern, which is the term of [Kashino, *et a*/., 1995b], where the interval between at least two simultaneous notes is a perfect fifth.

We defined the recognition rate, *R,* as

$$R = 100 \cdot \left( \frac{right - wrong}{total} \cdot \frac{1}{2} + \frac{1}{2} \right) \quad [\%],$$

where *right* is the number of correctly identified and correctly source-separated notes, *wrong* is the number of spuriously recognized (surplus) notes and incorrectly identified notes, and *total* is the number of notes in the input; this is the same definition as in the above mentioned papers. From preliminary experiments, number of taps of the FIR filter was chosen to be 40 for the template-adaptation-on condition. The template-adaptation-off means that the number of taps of the FIR filter was fixed to 1.

In this test, if one use the same waveforms as the templates as the ones used for test signals, these waveforms will completely match and results will become in-

Table 1: Results of a benchmark test indicating that both kinds of adaptive processing discussed in Section 2 have improved the note recognition accuracy.

| | | Template Filtering | |
|---|---|---|---|
| | | On | Off |
| Phase Tracking | On | 77.3 % ± 4.1 % | 64.7 % ± 4.9 % |
| | Off | 61.0 % ± 4.8 % | 57.8 % ± 4.8 % |

± : 95 % confidence intervals

Table 2: Results of a music recognition test.

| | | Template Filtering | |
|---|---|---|---|
| | | On | Off |
| Phase Tracking | On | 66.3 % | 61.0 % |
| | Off | 52.7 % | 52.3 % |

appropriate. Therefore we used different manufacturers' instruments, for example, Boesendolfer's piano and Yamaha's piano, in making test signals and templates, respectively.

The results are listed in Table 1, which clearly show advantages of the processing scheme developed in Section 2. The condition where both template adaptation and phase tracking are turned off is equivalent to the matched filtering, which is a conventional signal processing method for signal identification.

### 4.2 Music Recognition Tests

We have evaluated the system using music sound signals. Table 2 lists the note recognition rates for a sample music: a live recording of a chamber ensemble "Auld Lang Syne", arranged in three parts and performed by violin, flute, and piano. A part of output of the agents network and a part of score-like data produced by the system are shown in Figures 6 and 7.

## 5 Related Work and Discussions

For the acoustic signal separation task, much work has been done since as early as 1970's. The approach using microphone arrays has been one of major research streams [Mitchell *et* a/., 197l] [Bell *et* a/., 1995], and the harmonic selection is another major method [Parsons, 1976] [Nehorai and Porat, 1986]. These approaches have been principally based on a single cue (localization of sources or harmonicity). On the other hand, works trying to integrate multiple cues for stream segregation are

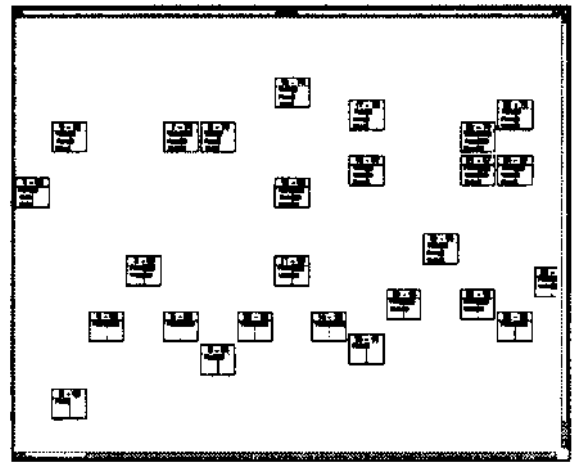

Figure 6: A part of output from agents. Ordinate: pitch and abscissa:time. Each square stands for a recognized note. Bars in each square denote the average powers of the activated agents.

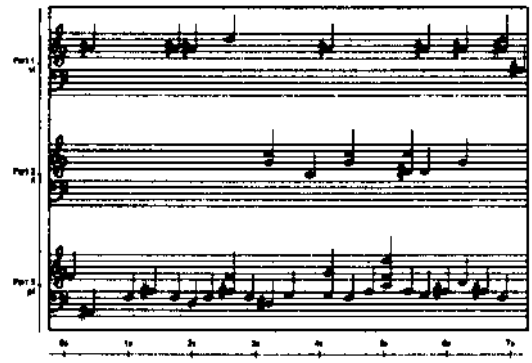

Figure 7: A part of results yielded by the implemented system. A note value recognition (e.g. quarter notes, half notes, ...) is not considered here and all the notes are displayed as quarter notes with a real-time scale.

recently emerging. The most closely related works to the presented architecture are the IPUS project [Lesser *et* a/., 1993], Nakatani *et al.* [1995], Kashino *et al.* [1995a] and Ellis [1996].

The IPUS is an acoustic signal understanding project based on the blackboard architecture[Lesser *et* a/., 1993], seeking adaptive processing according to the input to the system. IPUS realized its adaptivity basically in a rule-based strategy while the Ipanema architecture does not employ symbolic rules for adaptation.

Nakatani *et al.* invented a speech segregation system that consists of multiple processing modules called a generator and tracers [Nakatani, *et* a/., 1995]. The function of the tracer is to trace harmonic structure, which is similar to the function that the promoter in our architecture performs. However, their system does not have an explicit mechanism to identify sound sources; it only

segregates harmonic or localized sound into signals. Our main point is the adaptive processing to trace the specific sound sources, investigating what the sources are.

Ellis proposed a prediction-driven architecture for an auditory scene analysis system [Ellis, 1996], where context sensitivity in scene interpretation is realized. In our current implementation of Ipanema, musical context does not affect the template adaptation scheme; the context is utilized in the information integrator.

The studies toward music recognition includes [Mont-Reynaud, 1985], [Chafe *et a*/., 1985] and [Brown and Cooke, 1994]. However, automatic music transcription systems or music stream segregation systems which can deal with given ensemble music played by multiple musical instruments with a reasonable accuracy have not yet been realized. A quantitative architecture was proposed by Kashino *et al.* [1995a], in which a Bayesian probability scheme was applied. However, their architecture does not yet include adaptive processing and has been applied only to artificial performances synthesized by a sampler [Kashino, *et ai,* 1995a]. The system presented here, on the other hand, was designed for, and tested by, real performances rather than sampler performances.

## 6 Conclusion

We have presented a new system architecture designed for auditory stream segregation. To cope with a variety of sounds that appear in the real world, two mechanisms, template filtering and phase tracking, have been devised. In implementation, we have taken advantage of modularity and scalability of the multi-agents approach. The adaptivities realized in this paper enable each agent to optimally, in the least-squares sense, track the individual sound included in input sound signals.

As an example application of the proposed architecture, we have built a music recognition system that recognizes instrument names and pitches of the notes included in ensemble music performances. Experimental results show that the adaptive mechanisms significantly improve the recognition accuracy in comparison to the matched-filter-based processing, which is a conventional signal detection/separation method.

This paper has focused on the adaptive mechanisms and left the information integrator, the post-processing module, almost untouched. However, information integration is an important issue to be addressed: our preliminary tests have shown that integration of the statistical information of note transitions improves the note recognition rate up to approximately 75 % in the same music recognition test as used here [Kashino and Murase, 1997]. To obtain further accuracy, we anticipate that sound source models that explicitly model variations of the sources would be necessary.

## Acknowledgments

The authors would like to thank Hiroshi G. Okuno, Takeshi Kawabata, and Tomohiro Nakatani for contributing valuable discussions with the authors. The authors also wish to thank Ken'ichiro Ishii for his support as the Executive Manager of their research laboratory.

## References

[Bell *et ai,* 1995] Bell A. J. and Sejnowski T. J.: An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation,* vol.7, 1129-1159, 1971.

[Bregman, 1990] Bregman A. S.: *Auditory Scene Analysis.* MIT Press, 1990.

[Brown and Cooke, 1994] Brown G. J. and Cooke M.: Perceptual Grouping of Musical Sounds: A Computational Model. *J. of New Music Research,* 23(1):107-132, 1994.

[Chafe *et ai,* 1985] Chafe C, Kashima J., Mont-Reynaud B., and Smith J.: Techniques for Note Identification in Polyphonic Music. In *Proc. of the 1985 Intl. Computer Music Conf.,* pp.399-405, 1985.

[Ellis, 1996] Ellis D.: *Prediction-driven Computational Auditory Scene Analysis.* PhD. Thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., 1996.

[Kashino, *et ai,* 1995a] Kashino K., Nakadai K., Kinoshita T., and Tanaka H.: Organization of Hierarchical Perceptual Sounds. In *Proc. of the 14th Intl. Joint Conf. Artificial Intelligence,* Vol.1, 158-164, 1995.

[Kashino, *et ai,* 1995b] Kashino K., Nakadai K., Kinoshita T., and Tanaka H.: Application of Bayesian Probability Network to Music Scene Analysis. In *Working Notes of the Computational Auditory Scene Analysis Workshop, IJCAI-95,* pp.32-40, 1995.

[Kashino and Murase, 1997] Kashino K. and Murase H.: Sound Source Identification for Music Based on Melody Extraction. In *Working Notes of the Computational Auditory Scene Analysis Workshop, IJCAI-97,* 1997.

[Lesser *et ai,* 1993] Lesser V., Nawab S. H., Gallastegi I., and Klassner F.: IPUS: An Architecture for Integrated Signal Processing and Signal Interpretation in Complex Environments. In *Proc. of the 11th National Conf. on Artificial Intelligence,* pp.249-255, 1993.

[Mitchell *et ai,* 197l] Mitchell O. M. E., Ross C. A., and Yates G. H.: Signal Processing for a Cocktail Party Effect. *J. Acoust. Soc. Am.,* 50(2):656-660, 1971.

[Mont-Reynaud, 1985] Mont-Reynaud B.: Problem-Solving Strategies in a Music Transcription System. In *Proc. of the Intl. Joint Conf on Artificial Intelligence,* pp.916-918, 1985.

[Nakatani, *et al,* 1995] Nakatani T., Okuno G. H., and Kawabata T.: Residue-Driven Architecture for Computational Auditory Scene Analysis. In *Proc. of the 14th Intl. Joint Conf. Artificial Intelligence,* Vol.1, 165-172, 1995.

[Nehorai and Porat, 1986] Nehorai A. and Porat B.: Adaptive Comb Filtering for Harmonic Signal Enhancement. *IEEE Trans, on ASSP,* 34(5):1124-1138, 1986.

[Parsons, 1976] Parsons T. W.: Separation of Speech from Interfering Speech by Means of Harmonic Selection. *J. Acoust. Soc. Am.,* 60(4):911-918, 1976.

[Pearl, 1986] Pearl J.: Fusion, Propagation, and Structuring in Belief Networks. *Artificial Intelligence,* 29(3):241-288, 1986.

# NEURAL NETWORKS

## Neural Nets 4:
## *Learning* Algorithms and Architectures