論文 / 著書情報
Article / Book Information

論題(和文)	WFSTに基づくＴ３音声認識デコーダ
Title(English)	
著者(和文)	大西 翼, Dixon Paul R., 古井 貞熙
Authors(English)	Oonishi Tasuku, Paul Dixon, SADAOKI FURUI
出典(和文)	情報処理, Vol. 51, No. 11, pp. 1440-1448
Citation(English)	, Vol. 51, No. 11, pp. 1440-1448
発行日 / Pub. date	2010, 11
権利情報 / Copyright	

8 WFSTに基づく T³音声認識デコーダ

大西　翼*　Dixon Paul R.**　古井 貞熙*
*東京工業大学　**(独) 情報通信研究機構

WFSTを利用した音声認識デコーダ開発の背景と目的

情報処理技術，および音声認識技術の発展により，大規模なデータを利用した音声認識が実現可能となってきた．今日の音声認識技術は，機械学習に基づいており，学習データが大規模であればあるほど，複雑なモデルを高精度に学習することができる．これにより話者や言葉の違いに頑健な音声認識を実現することが可能となる．また，大規模なデータを背景としたさまざまなモデリングにより，自由度の高い音声認識を行うことができる．これに伴い，音声認識を行うエンジンであるデコーダには，モデルの複雑化や多様化への対応が求められる．

近年，重み付き有限状態トランスデューサ(WFST)を利用した音声認識手法[1]が提案され，次世代の音声認識デコーダの枠組みとして大きな注目を集めている．WFSTは数学的に簡明な枠組みで，個々のWFSTの合成や最適化を自動で行うことができるという特徴がある．WFSTを利用した音声認識手法では，従来の認識時に探索ネットワークを動的に構築するアプローチ（動的な探索ネットワーク展開のアプローチ）とは異なり，認識に先立ち探索ネットワークを事前に構築するアプローチ（静的な探索ネットワーク展開のアプローチ）を用いている．これにより，WFSTを利用した音声認識デコーダは，従来の音声認識デコーダと比べて，以下の3つの利点を持つ．

システムの保守・拡張性：探索ネットワークは，デコーダと独立に構築される．そのため従来のデコーダで必要だった探索ネットワーク構築のための複雑な処理が不要となり，デコーダの保守・拡張性が向上する．

モデルの利用に対する汎用性：認識で利用する個々のモデル（単語発音辞書，言語モデルなど）はWFSTの形式で表現される．そのため，これらを組み合わせた探索ネットワークはWFST上の演算[1]を用いるだけで自動的に構築することができる．これにより，さまざまなモデルを組み合わせた複雑な認識処理を容易に実現できる．

計算効率性：探索ネットワークの最適化はWFST上の演算を利用するだけで自動的に実現できるため，常に計算効率の高い音声認識を実現できる．

一方，WFSTに基づく音声認識手法は，静的な探索ネットワーク展開のアプローチに基づいているため，探索ネットワークのすべての状態を事前に展開する必要がある．このため認識時の動的なモデルの変化への対応の困難さや消費メモリ量の増大などの課題がある．

我々は，次世代の音声認識デコーダ実現のため，経済産業省の開発プロジェクトの支援を受けて，WFSTを利用した音声認識デコーダ：Tokyo Tech Transducer-based Decoder（T³ Decoder：ティーキューブドデコーダと発音）を開発した[2]．本デコーダは，実用的な音声認識アプリケーションの開発や最先端の音声認識研究を支援するためのさまざまな機能が実装されている．本稿では，これらの機能の中で，特徴的な以下の3つの機能について解説する．

最適化付き on-the-fly 合成：WFST音声認識で課題となるモデルの動的な変化への対応やメモリ消費量の増大に対処するため，動的に探索ネットワークを合成する「最適化付き on-the-fly 合成」を実装している．これにより，モデルの

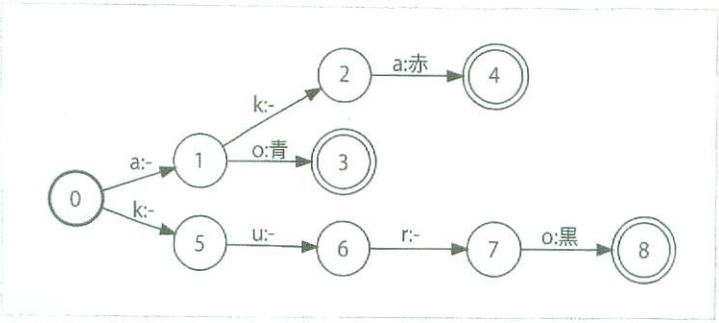

図-1 WFST L の例

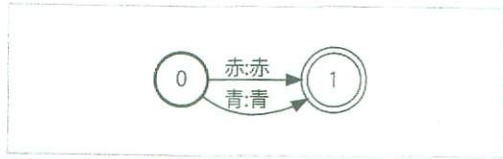

図-2 WFST G の例

適応化による頑健な音声認識の実現や数十万から数百万語彙規模の超大語彙認識による語彙外単語(未知語)の問題の軽減を可能とする.

Graphics Processor Unit (GPU) を利用した音響尤度計算：実用的な時間で大規模な音響モデルと入力音声のマッチング処理を行うため，一般的なPCに搭載されているGPUを用いることによる音響尤度計算機能を実装している．これにより廉価なPCでも，高精度・高速な音声認識を実現可能にしている.

Voice Activity Detection (VAD) の信頼度を利用した音声認識手法：従来の音声・非音声を判定するVADの枠組みは，フロントエンドで計算された音声検出情報がデコーディングに利用されず，これによる情報欠落が雑音環境下における認識精度低下の1つの原因となっていた．そこでVADの信頼度をデコーディングで利用する機能を実装し，雑音の大きい環境下における頑健な音声認識を実現した．これにより野外など騒音の大きい環境におけるデコーダの利用を可能にしている.

以下では，WFSTを利用した音声認識およびT³デコーダの詳細と，これらの技術の詳細について述べる.

WFST音声認識

WFSTは与えられた入力記号列に対して，状態遷移を繰り返すことで，記号列と重みを出力する有限状態オートマトンの一種である．これは記号列を入力とし記号列と重みを出力する変換器とみなすことができる．WFSTには，1つまたは複数のWFSTを受け取り，ある性質を持ったWFSTを生成する演算が定義されている．たとえば，合成演算[1]は，2つのWFST T_1, T_2 を受け取り，T_1 と T_2 の直列の変換を行うWFSTを出力する.

WFSTに基づく音声認識では，まず単語発音辞書，言語モデルなど，認識で利用する構成要素を各々WFSTの形式で表現する．図-1，図-2に，単語発音辞書，言語モデルを表現したWFSTの例を示す．図の各ノードが状態を表し，太線で囲まれた状態が初期状態，二重線で囲まれた状態が最終状態を表す．アークは状態遷移を表し，入力シンボル，出力シンボルが，“入力シンボル:出力シンボル”と表記されている．出力シンボル中の“-”はシンボルの出力をせずに遷移を行うことを表す．図-1は，単語の発音を表すWFST L で，音素列“ao”，“aka”，“kuro”から，その音素列に対応する単語“青”，“赤”，“黒”に変換する．図-2は，言語モデルを表すWFST G で，単語“青”，“赤”のみを受け付ける文法モデル[☆1]となっている.

一般的な音声認識では，構成要素として以下の3つのWFSTを用いる.

C：文脈依存音素から文脈非依存音素へ変換するWFST(音素の変動を詳細に表現するモデル)

L：文脈非依存音素から単語列へ変換するWFST（単語の発音を定義するモデル）

G：単語列から言語モデルに従う単語列へ変換

☆1 単語の出現規則を表現したモデル.

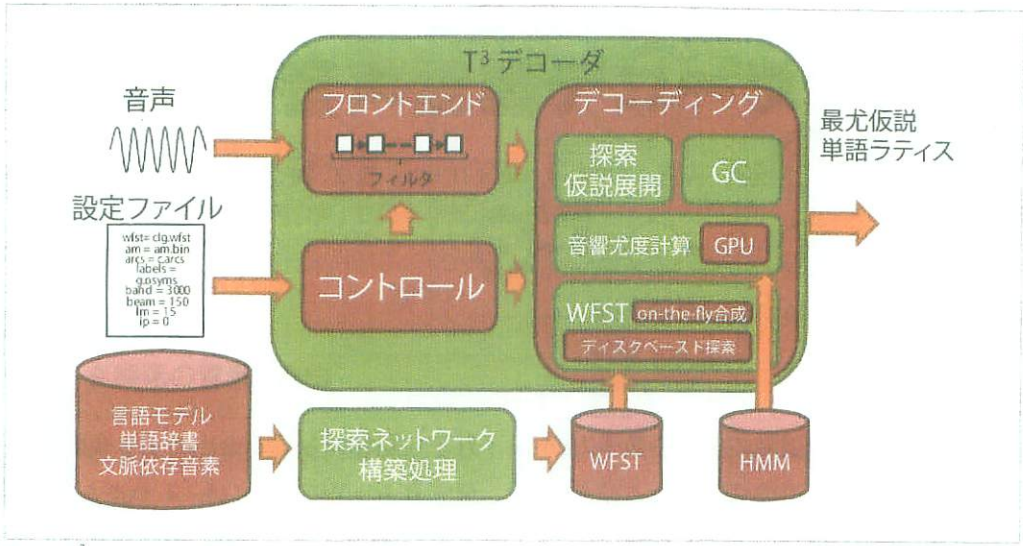

図-3　T³ デコーダの構成

する WFST（単語の生成規則や単語間のつながりやすさを表現するモデル）

次に，これらの構成要素を合成演算[1]により 1 つの WFST に合成し，決定化[1]，最小化[1] などの最適化演算により WFST の最適化を行う．合成，決定化，最小化演算をそれぞれ∘，*det*，*min* で表現すると，探索ネットワークは以下の式により構成される．

$$min(det(C \circ det(L \circ G))) \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (1)$$

WFST に基づく音声認識デコーダは，式(1)により生成された WFST を探索ネットワークとして利用する．音声認識は，事前に構築された探索ネットワーク上の最短(コスト最小)パスを探索することで実現する．

音声認識で利用するモデルは WFST の形式で表現しさえすれば，WFST 上の演算を行うことにより自動的に探索ネットワークに組み入れることができる．このため文法モデルや N-gram モデル（単語連鎖統計モデル）など，利用するモデルの種類を増やすためにデコーダのプログラムを拡張する必要がない．これによりさまざまな種類のモデルを利用した音声認識処理を容易に実現できる．

また，探索ネットワークの最適化は，WFST 上のさまざまな最適化演算を施すだけで自動的に実現できる．さらに探索ネットワークは，認識前に構築されるため，認識時に探索ネットワークを構築するオーバーヘッドを削減することができる．これらの特徴から，従来の音声認識手法と比べて，計算効率の高い音声認識を実現できる．

一方，WFST に基づく音声認識手法は，静的な探索ネットワーク展開のアプローチに基づいているため，認識に先立ち探索ネットワークのすべての状態を展開する必要がある．このため，認識に利用するモデルが変更された場合には，式 (1) に基づき，探索ネットワークを再構築する必要がある．このことはモデルの適応など，認識時にモデルが動的に変化する処理への対応を困難にする．また，探索ネットワークのサイズは，構成されるモデルのサイズに対して，組合せ的に大きくなる．このため超大語彙など巨大なモデルを扱う場合にはメモリ消費量が大きくなり過ぎ，静的に探索ネットワークを展開することが困難になる．これらの問題に対処するため，認識時に合成演算を行い動的に探索ネットワークを構築する「on-the-fly 合成」が提案されている．これにより，「動的な探索ネットワーク展開のアプローチ」と「静的な探索ネットワーク展開のアプローチ」の両方の利点を持った音声認識処理が可能になる．

T³ デコーダの特徴

図-3 に T³ デコーダの構成を示す．デコーダは大

きく“コントロール”，“フロントエンド”，“デコーディング”のユニットに分けられる．コントロールユニットでは，設定ファイルに記述された情報を基にフロントエンドでの変換処理の決定やパラメータの設定などを行う．フロントエンドユニットでは，音声認識を行うためのフロントエンド処理を行う．デコーディングユニットでは，最尤仮説の探索，音響尤度計算などのデコーディング処理を行う．

●フロントエンド

本デコーダのフロントエンドは，音声認識デコーダ Sphinx[3] で用いられている多段フィルタによる変換処理方式を採用している．この方式では，フィルタと呼ばれる計算ユニットに順次データを通すことで，音声認識のフロントエンド処理を実現する．フィルタの処理や変換順序を適宜設定することで，利用目的に応じたフロントエンドを柔軟に実現できる．本デコーダでは，音声データから，代表的な特徴ベクトルである MFCC（メル周波数ケプストラム係数）に変換するためのフィルタセット（「窓掛け」や「FFT」などを行うフィルタ）や VAD を行うためのフィルタセットが実装されている．

●デコーディング

本デコーダの認識処理は，入力された音声（特徴ベクトル）に対する探索ネットワーク上の最短パスを求めるための探索処理，入力された音声と音素との類似度（尤度）を求める音響尤度計算，探索ネットワークを表現する WFST への状態および状態遷移情報へのデータアクセス，認識結果出力から構成される．以下では，これらの詳細について述べる．

《探索》

探索は，フレーム同期型の 1 パス方式で実行される．フレーム同期型では，各フレーム（ある一定の区間ごとに切り出された音声）の仮説の算出を一括して行う．たとえば，i フレーム目のすべての仮説は，i フレーム目の探索処理を行う際にすべて算出される．フレーム同期型の 1 パス方式は，実装が簡単であり，リアルタイムでのアプリケーションとの親和性も高いという利点がある．本デコーダでは，効率的な探索を行うため，第一位の仮説からの尤度差を用いた枝刈りや保持仮説数の上限値を用いた枝刈りを行う．

《音響尤度計算》

音響尤度は，音素の音響的なモデル化として一般的に用いられる混合ガウス分布モデル（GMM）（ガウス分布の重み付き和による確率分布）により計算される．高速化のためガウス分布の足切り計算（Gaussian pruning）手法や GPU を利用した音響尤度計算手法を行う．

《WFST へのデータアクセス》

通常，WFST の状態データは，メモリ上に一括して保持される．この方式は状態データに高速にアクセスできるという利点があるが，すべての状態をメモリ上に保持する必要があるため，認識時の消費メモリが大きいという欠点がある．省メモリ化対策として，ディスクベースド探索ネットワークや on-the-fly 合成を実装している．

ディスクベースド探索ネットワークでは，探索ネットワークをあらかじめディスク上に展開し，認識時に必要な状態データだけをディスクからメモリに読み込み利用する．そして状態データを利用し終わると，それをメモリから解放することで，メモリ消費量を抑える．この方法は，探索ネットワークの一部のデータのみをメモリ上に保持するため，大幅に消費メモリ量を削減することができる．しかし探索ネットワークが巨大であり，あらかじめディスク上に展開することが難しい場合や探索に利用する WFST が動的に変化する場合には利用することができない．

On-the-fly 合成では，認識時に WFST の合成演算を行うことで，探索ネットワークを動的に合成する．認識に必要な状態のみを合成することで，認識時の消費メモリ量を抑えることができる．ディスクベースド探索ネットワークとは違い，探索ネットワーク全体をあらかじめ合成する必要がないので，より巨大な探索ネットワークを利用した音声認識やモデルの適応化処理などが可能となる．しかし，動的

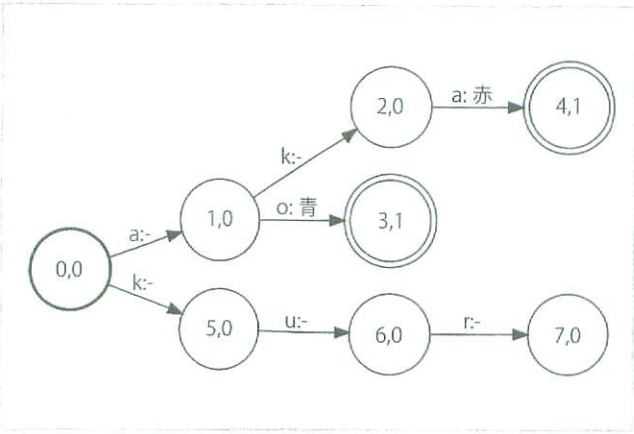

図 -4　合成後の WFST $L \circ G$

な合成演算を行う必要があるため，ディスクベースド探索ネットワークと比べて，状態データにアクセスするためのオーバーヘッドが大きくなる．

《認識結果出力》

本デコーダは，音声検索やリスコアリング処理を利用したアプリケーションとの親和性を高めるため，最尤仮説のほかに下位候補の出力を行うことができる．また，字幕付与などのリアルタイムのアプリケーションに対応するため，発話の途中で早期に単語列を出力する逐次デコーディングが実装されている．

最適化付き on-the-fly 合成

● on-the-fly 合成

WFST の合成演算では，ある状態を合成するときに他の状態と独立して合成することができる．そのため，探索に必要な状態だけを合成する「on-the-fly 合成」が可能となる．on-the-fly 合成により音声認識を行った場合，部分的に探索ネットワークを合成することができるので，メモリ消費量を削減することができる．一方で，合成された探索ネットワークは，最適化が施されていないため，探索効率が低下する．そこで本デコーダでは on-the-fly 合成と同時に WFST の最適化を行う「最適化付き on-the-fly 合成」を行う．これにより，on-the-fly 合成を用いた音声認識の高速化を実現している．

● on-the-fly 合成時の WFST の最適化

本デコーダでは，on-the-fly 合成時の最適化処理として，最終状態に到達しない状態（デッドエンド状態）の合成を回避する「デッドエンド状態の回避処理」，重みの先読みを行う「ダイナミックプッシング」を行う（詳しい説明については文献 4）を参照）．以下では，最適化処理の 1 つである「デッドエンド状態の回避処理」の概要を述べる．

図 -1，図 -2 の WFST を合成演算 [1] により合成した WFST $L \circ G$ を図 -4 に示す．合成された WFST の各状態は，L と G の状態の組により表記される．たとえば，“1,0” は，L の状態 “1” および G の状態 “0” から合成された状態であることを表す．

合成演算を行った場合，図 -4 の状態 “7,0” のように，遷移する先の状態が 1 つも存在しない非最終状態（デッドエンド状態）を生成する場合がある．デッドエンド状態および，それにしか到達しない状態（図の状態 “5,0”，“6,0”）は，最終状態への最短パスの探索には不要となるため，そのような状態を合成しないことが探索効率上望ましい．

このため本デコーダでは，WFST L 上の出力シンボルの先読みを行うことで，合成時に発生する無駄な状態の合成を回避している．たとえば，状態 “5,0” を合成する場合，シンボルの先読みを行うことで，L の状態 “5” からは，シンボル “黒” が出力され，G の状態 “0” からは，シンボル “赤” または “青” が入力として受け入れられることが分かる．L の状態から先読みされる出力シンボルの集合（先読みシンボル集合）と G の状態から入力として受け入れられるシンボルの集合の共通集合が空集合である場合には，デッドエンド状態に到達すると判定することができる．これから状態 “5,0” の合成を回避する．本デコーダでは，このような判定を行うことで，デッドエンド状態の合成を回避している．

GPU を利用した音響尤度計算の高速化

混合ガウス分布を音響モデルとして利用する音声認識では，ガウス分布の混合数の増加に伴い，音響

尤度計算に多くの時間を必要とする．このため効率的な音響尤度計算は，高速な音声認識を実現するために非常に重要である．そこで本デコーダでは，GPUが持つ高速な行列演算能力を利用した音響尤度計算の高速化手法を実装している．

ガウス分布による音響尤度は，行列演算の積により計算することができる[5]．GPUを用いた音響尤度計算手法では，この積の演算をGPU上で計算することで，GPUが持つ高い計算能力を利用する．

CPUを用いた音響尤度計算では，探索に必要な音素の音響尤度を随時計算する．一方，GPUを用いた音響尤度計算手法では，すべての音素の音響尤度を一括して計算する．また，フレームごとに音響尤度を計算するのではなく，Nフレームまとめて同時に計算するバッチ処理を行う．これらの処理により，オーバーヘッドとなるCPUとGPU間のデータ通信時間を削減する．さらに，GPUを用いた音響尤度計算手法では，すべての音素の音響尤度を計算するため，音響尤度計算の処理を探索処理と独立に実行することができる．これにより探索を行っている間に，次のNフレーム分の音響尤度計算を並行して計算する並列処理が可能となり，音声認識のさらなる高速化が実現できる．

VADの信頼度を利用した音声認識手法

認識システムに入力された音を音声・非音声に判定するVADの技術は，実用的な音声認識システムの基盤技術である．一般的なVADの実装方式では，フロントエンドで入力された区間を音声または非音声に判定し，音声と判定された区間を後段のデコーダに渡す．逆に，非音声と判定された区間は，フロントエンドで棄却し，認識に利用しない．

フロントエンドでVADを行う実装方式（フロントエンド型VAD）の問題として「音声・非音声を確定的に判定する点」と「音声区間の検出情報を認識に利用しない点」がある．雑音環境下では，音声・非音声を確定的に判定した場合，音声区間を棄却する誤りが増加する．一度棄却された音声区間は，デコーダで復元することができないため，音声区間の誤棄却の増加は，認識率低下の大きな原因となる．また，音声区間の検出情報は，入力区間の棄却判定にのみ利用される．そのため，相対的に高い確率で音声と判定された区間も低い確率で音声と判定された区間も同じ音声区間としてデコーダで認識される．雑音環境下など音響モデルと入力環境との乖離が大きい場合には，高い確率で音声と判定された区間であっても，デコーダが無音として誤認識する場合がある．

本デコーダでは，これらの問題を解決するためVADの信頼度をデコーディング時の仮説スコアの調整に利用する手法[6]を実装している．この手法では，音声・非音声をモデル化した各GMMにより計算された信頼度を仮説の音響尤度に加えることで，スコアの調整を行う．本手法では，入力された音をすべて認識するため，音声区間の誤棄却の問題が軽減できる．また，音声(非音声)の信頼度が高い区間では，音声(非音声)を表す仮説のスコアが高くなる．これにより，音声と判定された区間を無音として認識する誤りが軽減できる．

さらに本デコーダでは，多様な入力環境に対応するため，音声・非音声を表すGMMの音響適応を行う．実用的なシステムでは，認識前にデータを収集し，それを用いてモデルを適応するといったアプローチをとることができない場合がある．そのため，認識時に動的にGMMを適応するオンライン適応が必要となる．そこで本手法では，MAP推定による適応手法により，オンライン教師なしでGMMの適応を行う．GMMの適応データを選別するため，信頼度によるデータ選択を行う．これらの処理により，従来のVADの枠組みであるフロントエンド型VADを凌ぐ頑健な音声認識を実現し，雑音の大きい環境下における本デコーダの利用を可能にしている．

T³デコーダの性能評価

本章では，T³デコーダに実装されている機能に

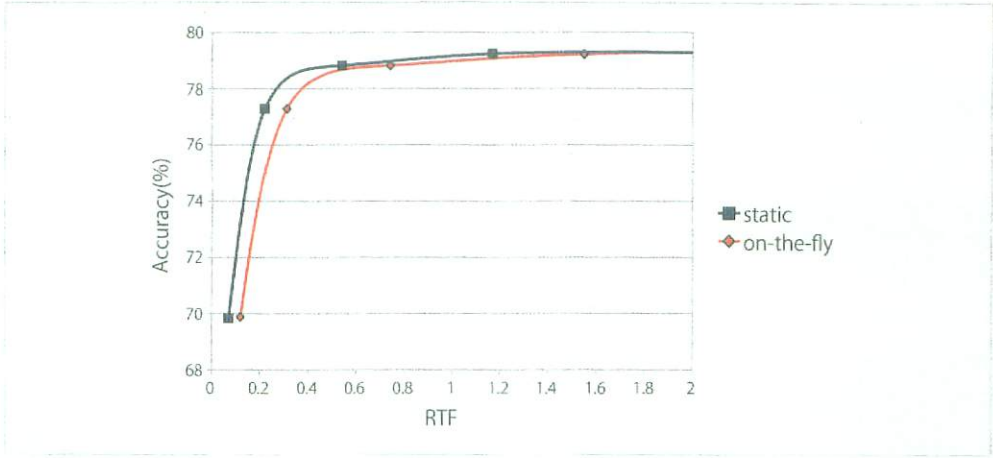

図-5　on-the-fly 合成の評価

ついての性能評価を行う．

●最適化付き on-the-fly 合成の評価

学習データとして日本語話し言葉コーパスを用い，評価データとして広く用いられている「テストセット 1」の男性 10 講演を用いた．詳しい実験条件は文献 2) と同様である．評価には，3.0GHz CPU (Intel Core2 Quad)，4GB メモリの計算機を用いた．静的に探索ネットワークを構築する場合は，式 (1) による WFST を用いた．on-the-fly 合成を行う場合には，$det(C \circ L)$ と G の WFST を用いた．

図-5 の "static" が静的に探索ネットワークを構築した場合，"on-the-fly" が on-the-fly 合成により動的に探索ネットワークを構築した場合の認識結果である．図の縦軸が単語正解精度，横軸が RTF（実時間比）である．探索を行う際には，仮説数の上限による枝刈りと，仮説の尤度差による枝刈り（ビームの幅）の 2 つのパラメータを用いているが，各点は仮説数の上限による枝刈りのパラメータを固定し，ビーム幅のパラメータを変化させることでプロットしている．

図から on-the-fly 合成を用いた場合，静的にネットワークを合成した場合と比べて，20 ～ 40% 程度の認識時間の増加が見られるものの，実時間で単語正解精度の収束点に到達していることが分かる．これから我々の提案する手法を用いることで，大きな認識速度の低下なく on-the-fly 合成が実現できることが分かる．

● GPU を利用した音響尤度計算の高速化の評価

実験条件は文献 2) と同様である．評価用計算機として 2.4GHz CPU (Intel Core2 Duo)，2GB メモリの計算機，グラフィックカードとしては NVIDIA 8800GTX (128 core G80 GPU) を用いた．

図-6 に 4 ～ 512 の混合数における認識時間と単語正解精度との関係を示す．図の縦軸が単語正解精度，横軸が RTF である．図から混合数の増加に伴う認識時間の増加が，ほとんど見られないことが分かる．これから，音響尤度計算に関する計算量のほとんどが削減されていることが分かる．また，非常に大きな混合数を用いた場合でも，実時間での音声認識が実現できることが分かる．これから GPU を用いた音響尤度計算手法を用いることで，複雑な音響モデルを用いた高精度な音声認識を実時間で実現できることが確認できた．

● VAD の信頼度を利用した音声認識手法

評価用データに，Drivers' Japanese Speech Corpus in a Car Environment (DJSC) の高速道路走行におけるハンズフリーコマンド発話を用いた．これは音声認識によるカーナビゲーションの利用を想定し作成されたコーパスで，自動車走行中にカーナビゲーションを音声で操作するために発声されたコマンド発話を収録している．S/N 比が -8 ～

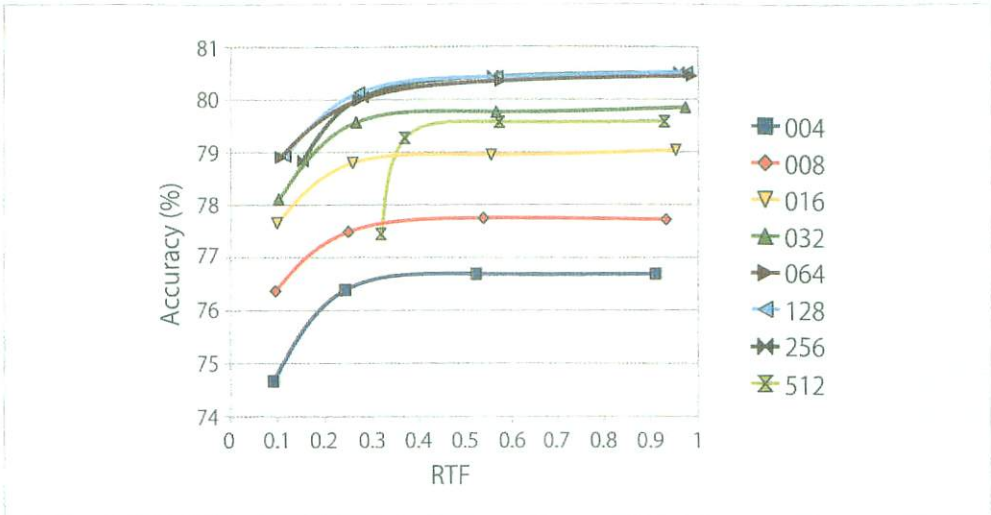

図-6 GPUを用いた音響尤度計算高速化の効果

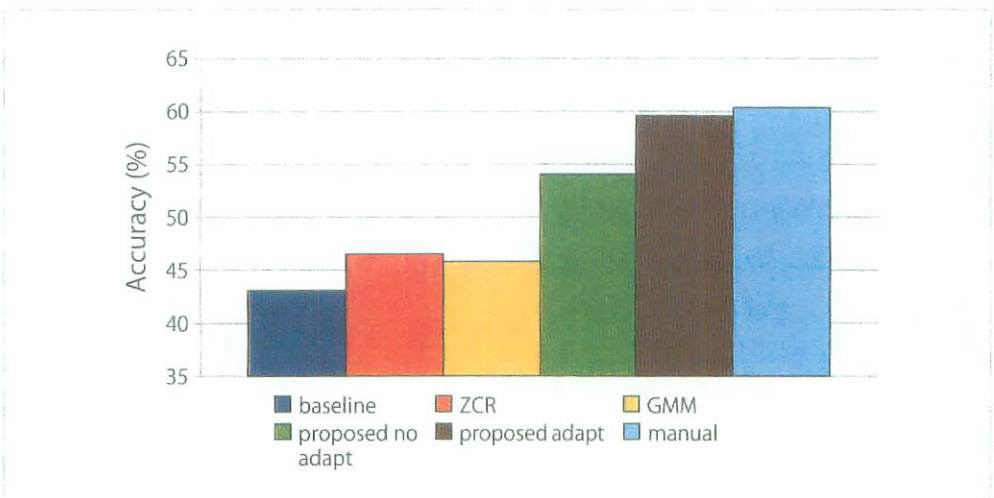

図-7 フロントエンドでVADを行った場合と提案手法の比較

10dBの高雑音環境の音声である．詳しい実験条件は文献6）と同様である．比較として，パワー・零交差数を利用するVAD手法と，音声・非音声のGMMの尤度比を利用するVAD手法を用いた．

図-7にパワー・零交差数およびGMMの尤度比によるVADをフロントエンドで行った場合，および提案手法を利用した場合の単語正解精度を示す．図のグラフと実験条件の関係は以下の通りである．

baseline：VADを行っていない場合

ZCR：パワー・零交差数によるVADをフロントエンドで行った場合

GMM：音声・非音声GMMの尤度比によるVADをフロントエンドで行った場合

proposed no adapt：提案手法でGMMのオンライン教師なし適応を併用しない場合

proposed adapt：提案手法でGMMのオンライン教師なし適応を併用した場合

manual：人手で付けられた情報を基に，フロントエンドで音声・非音声の判別を行った場合

なお各手法のパラメータは，テストセット全体に対して最適な値を人手により設定している．図からVADを行わない場合の単語正解精度は43.1%，パワー・零交差数を用いた場合は46.5%，GMMの尤度比を用いた場合は45.8%であった．一方，提案手法の単語正解精度は54.8%であり，フロントエンドでVADを行った場合と比べて，大幅な単語正

解精度の改善が得られた．これから，提案手法では従来のVADの方式を上回る認識精度が得られることが分かった．さらに，オンライン適応を組み合わせることで59.6%と大幅な単語正解精度の改善が得られ，人手による音声・非音声の切り出しを行った場合と，ほぼ同程度の単語正解精度が得られた．以上の結果から本手法の有効性が確認できた．

現状と今後の指針

本稿では，東京工業大学で開発を進めているT^3デコーダの概要と性能評価結果を述べた．性能評価実験では，最適化付きon-the-fly合成手法，GPUを利用した音響尤度計算手法，VADの信頼度を利用した音声認識手法を評価した．これから，on-the-fly合成時に探索ネットワークの最適化を同時に行うことで，静的に探索ネットワークを構築した場合と比べて，大きな認識速度の低下なく実時間で認識可能であることを確認した．また，GPUを音響尤度計算に用いることで，複雑かつ高精度な音響モデルを利用した場合でも実時間で認識可能であることを確認した．また，VADの信頼度を利用した音声認識手法を用いることで，高雑音の実環境タスクにおいて，高い頑健性を実現できることを確認した．これらの実験から数万語規模の大語彙連続音声認識で十分な認識性能が得られること，高雑音環境下で頑健に音声認識が実現できることを確認した．

今後の指針として，大規模なデータから学習された複雑なモデルを利用した音声認識の実時間処理および実用化アプリケーションとの連携強化が挙げられる．近年，超大語彙連続音声認識や複数の言語モデルを利用した音声認識など，認識モデルのさらなる複雑化が進んでいる．このようなモデルにおける実時間での音声認識を目指していきたい．また，実用化アプリケーションのためのプラットフォームの開発，耐雑音性の向上など，本デコーダを利用したアプリケーション開発のための機能強化を行っていきたい．

なお本デコーダは，情報通信研究機構（NICT）の高度言語情報融合（ALAGIN）フォーラムから，近く公開される予定である．

参考文献

1) Mohri, M., Pereira, F. and Riley, M. : Weighted Finite-state Transducers in Speech Recognition. *Computer Speech and Language*, Vol.16, No.1, pp.69-88 (2002).
2) Dixon, P., Oonishi,T., Iwano, K. and Furui, S. : Recent Development of Wfst-based Speech Recognition Decoder, In *Proc. Asia-Pacific Signal and Information Processing Association 2009 Annual Summit and Conference*, pp.138-147 (2009).
3) Lamere, P., Kwok, P., Walker, W., Gouva, E., Singh, R., Raj, B. and Wolf, P. : Design of the CMU Sphinx-4 Decoder In *EUROSPEECH*, pp.1181-1184 (2003).
4) 大西　翼，ディクソン・ポール，岩野公司，古井貞熙：WFST音声認識デコーダにおけるon-the-fly合成の最適化処理，電子情報通信学会論文誌（D），Vol.J92, No.7, pp.1026-1035 (2009).
5) Saraclar, M., Riley, M., Bocchieri, E. and Goffin, V. : Towards Automatic Closed Captioning : Low Latency Real Time Broadcast News Transcription, In *Proc. ICSLP*, pp.1741-1744 (2002).
6) 大西　翼，岩野公司，古井貞熙：音声・非音声の信頼度を利用した雑音に頑健な音声認識デコーダの検討，電子情報通信学会技術報告，Vol.110, No.81, pp.49-54 (2010).

（平成22年7月26日受付）

大西　翼　oonishi@furui.cs.titech.ac.jp
2006年東京工業大学工学部情報工学卒業．2008年同大学院修士課程修了．現在，同大学院博士課程在籍．

Dixon Paul R.　paul.r.dixon@gmail.com
2000年バーミンガム（英国）大学電子工学学科卒業．2007年バーミンガム（英国）大学電子工学博士卒業．2006～10年東京工業大学研究員．現在，情報通信研究機構研究員．

古井貞熙（正会員）furui@cs.titech.ac.jp
1970年東京大学大学院計数工学専攻修士課程修了．工博．NTTヒューマンインタフェース研究所音声情報研究部長，古井特別研究室長などを経て，1997年より東京工業大学教授．附属図書館長．音声情報処理の研究に従事．紫綬褒章，文部科学大臣表彰など．